UNIVERSITY OF LONDON THESIS

Degree PhD Year 2006 Name of Author DIMITRIOS TERZIS

AN ARCHITECTURE FOR

LARGE-SCALE IP TELEPHONY NETWORKS

Dimitris Terzis

A dissertation submitted in partial fulfilment

of the requirements of the degree of

Doctor of Philosophy

of the

University of London

Department of Computer Science

University College London

London, Wednesday 8 November 2006

To my family,

for their continuous support and care.

Abstract

After the explosion of Internet and World Wide Web use in the 1990s, the Internet Protocol has become the de facto standard for networked multimedia communications, sidestepping theoretically superior technologies like Frame Relay and ATM. This fact, combined with the higher efficiency of packet switching over circuit switching, has resulted in Voice over IP (VoIP) networks being increasingly preferred to the General Switched Telephone Network (GSTN), a trend that continues to accelerate in the 21st century. However, inherent technical difficulties related to the predominantly "best-effort" nature of IP, have restricted VoIP to the inside of corporate intranets or to the backbone of carrier networks, and no universal "IP Switched Telephone Network" (ISTN) exists to replace the GSTN.

This thesis investigates the applicability of the ISTN concept, on the assumption that the establishment of such an infrastructure will require enhanced scalability and tighter integration of technologies related to all three areas of end-to-end communication: the LAN (end users), for whom appropriate applications and usage models must be proposed, so that IP Telephony can be widely adopted; the MAN (access networks), particularly those interfacing with the GSTN, so that interoperability is secured; and the WAN (backbone), where call routing is expected to be a crucial factor in the deployment of a unified, worldwide VoIP infrastructure.

Examining the proposals currently used in the above three areas, reveals a number of shortcomings, which render the introduction of an alternative solution a necessity for ISTN to be realised in the near future. Therefore, a new architecture for large-scale IP Telephony is introduced and evaluated in this thesis, using improved designs for three core components, one per area addressed: a user agent for the LAN, a gateway for the MAN and a call routing mechanism for the WAN, which is generally more scalable than existing (and under-researched) ones. Finally, potential enhancements to the proposed architecture are suggested and certain directions for further investigation are identified.

Acknowledgements

This work owes a lot to the guidance and support of Professor Jon Crowcroft, now at Cambridge University, who not only was my supervisor until he left UCL, but continued to guide me ever since. My second supervisor at the same period, Dr. Vicky Hardman, has also significantly contributed to my knowledge of the rather vast subject of Voice over IP, with emphasis on speech coding and audio tools, so additional thanks are owed to her. During the writing up phase of this thesis, the help of my newly assigned supervisor, Dr. Stephen Hailes, has also been both substantial and crucial. Dr. Socrates Varakliotis has also provided several valuable comments, after reading earlier drafts of this document.

A number of other people have helped in forming the philosophy, background and choices related to the directions of this research, including former colleagues in three technology excellence centres: the British Telecom Labs (Adastral Park), at Martlesham Heath, in Suffolk, where I spent nearly 6 months during a placement; the Nortel Labs, at Harlow, in Essex, were I worked for 2.5 years, as the company funded a sizeable part of this work (the gateway) in an earlier contract; and the world-class Department of Computer Science of University College London (UCL-CS) itself, which has been at the cutting edge of networking research (and VoIP, in particular), for a number of decades, to date.

Finally, a lot is owed to two Greek foundations: the Bodossakis Foundation, which provided the initial means for me to seek postgraduate education in the UK by partially funding my work during its first 2 years, and to the Bakalas Foundation which also partially funded me for the next 3 years; their support was indispensable for completing the minor earnings I had from part-time commercial and industrial assignments during most of the long journey followed for the development of this PhD thesis.

London, Thursday 26 October 2006

Dimitris Terzis

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

The Internet Protocol (IP) has established itself as the preferred method of packet-based voice transportation since the 1990s, following continuous research and development efforts that can be traced back to the early 1970s [127]. The evolutionary period elapsed in the past 15 years has resulted in the emergence of a wide variety of IP Telephony products and services, for end users, network access and the backbone, currently including: open standards for signalling (mainly H.323 [188] and SIP [318]) and media (RTP [337]) transportation; proprietary but popular and effective control mechanisms, like the Cisco Skinny [220] and Skype [21]; sophisticated audio clients with telephone-like user interfaces ("softphones"), several of which are produced by vendors that also offer connectivity to telephony networks, thereby acting as non-incumbent operators [U23]; conferencing applications with additional support for new services, like instant messaging and presence (e.g., FWD.Communicator [U39], Gizmo [U12], Google Talk [U13], Jabber [U24], NetMeeting [U29], VoipBuster [U64], Windows Live Messenger [U67], Yahoo! Messenger [U68]); peer-to-peer VoIP networks (Skype [U48]); international VoIP providers (e.g., BroadVoice [U4], Vonage [U65]); a variety of broadband access technologies, such as xDSL, cable, Wi-Fi, 3rd Generation (3G) and 4th Generation (4G) mobile networks, that natively support VoIP and other intelligent features, like the Multimedia Messaging Service (MMS) [363], [390], [U54]; efficient Quality of Service (QoS) mechanisms like MPLS [310]; triple play (data, voice, video over IP) and quad play (data, voice, video, and mobile communication over IP) suppliers like Sky [U47] or HomeChoice [U14]; examples of an emerging convergence of fixed and mobile voice network technologies, such as the BT Fusion [U2], ultimately facilitated by technologies like the IP Multimedia Subsystem (IMS) [54]; protocols (like MGCP [5] and H.248/MEGACO [134], [187]) for controlling distributed gateway devices, in order to increase scalability; implementations of monolithic telephony switches in software (softswitches) [281]; and integrated multimedia network architectures, like the BT 21st Century Network (21CN) [U2], to mention just a few. Not surprisingly, the worldwide use of IP Telephony services is increasing rapidly and population coverage is expanding equally fast; in 2005, for instance, international VoIP traffic volumes collectively exceeded 36 billion minutes, thus contributing nearly 19% of the total (switched plus packet) voice traffic across the planet, while maintaining a strong growth potential [U53].

At the same time, a number of significant technical issues remain unresolved [70], [79], [128], [237], [269], [281], [332], [377]: the best-effort nature of IP continues to hamper reliability and Quality of Service (QoS); serious security threats such as eavesdropping (e.g., VOMIT [U63]), privacy intrusion (e.g., SPIT [305]) and denial of service attacks have emerged; the (hardware or software) user interface to VoIP services is far from standardised; interoperability with fixed and mobile telephony networks is partial; and no universal call routing mechanism has been developed yet. As a result, VoIP remains restricted inside the borders of organisational intranets or provider backbones and for the rest of the world, including most of the nearly 1 billion Internet users (estimated to reach 1.7 billion by 2010, as shown in Figure 1.1), the technology cannot, still, be offered as a viable alternative to the GSTN.

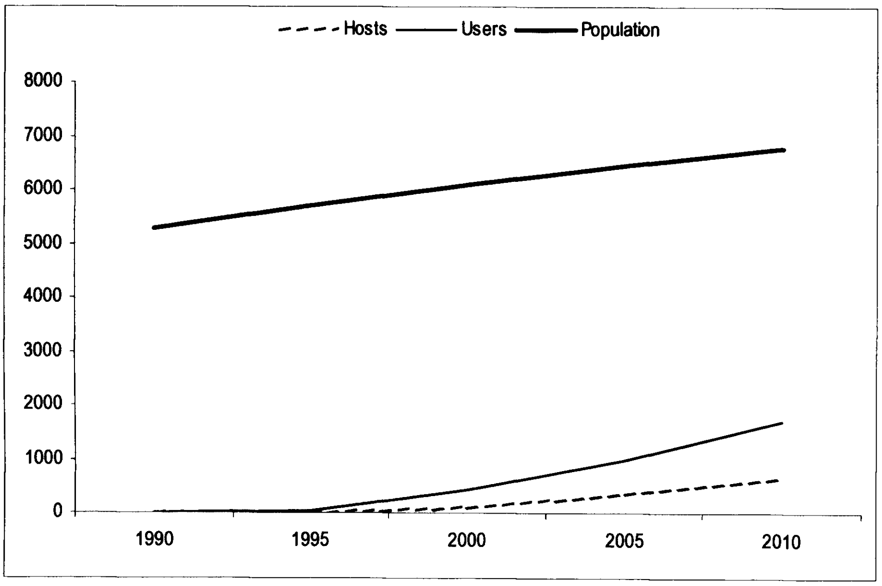


Figure 1.1: Number of Internet users worldwide

Despite these difficulties, the situation is gradually improving, with the spread of broadband networking (provided through xDSL, wireless communication or satellites), the development of IP QoS enhancement mechanisms (Integrated Services, Differentiated Services, MPLS), the adoption of VoIP for next generation (3G and 4G) mobile networks, the relaxation of service levels by non-incumbent fixed line and mobile operators, and the maturation of IP Telephony standards [253]. Thus, from a purely technological perspective, the deployment of VoIP services to a large number (hundreds of millions, or more) of end users is already feasible [128]. This deployment could ultimately lead to the unification of circuit-switched and packet-switched voice networks under the umbrella of IP, if GSTN-level quality and features are offered, to assist technology adoption [276], [309]. In fact, it is reasonable to assume that such a (rather social) requirement will be instrumental to the success of VoIP on a large scale.

Based on the above findings, this thesis investigates the problem of establishing a global IP Telephony network and proposes a new architecture that supports the "call anyone, anywhere, anytime" model of operation across heterogeneous (GSTN and IP-based) networks. The motivation for such a research, the analysis of the problem addressed, the hypothesis associated with its solution and the methodology followed to achieve it are presented in the rest of this chapter. A list of contributions and related work, as well as an overview of the structure of the thesis, are also included in the end.

1.2 Motivation

The main motivation for this research has been the combination of a long-term involvement with cutting edge VoIP technologies and the realisation that no comprehensive, end-to-end architecture for a unified, converged IP Telephony network exists to date. Significant achievements have been recorded during the past decade, but still without the full picture (i.e., a global-scale IP Telephony infrastructure) in mind. This is particularly evident in the software and hardware implementing the three core fields of communications technology, i.e. in the following three areas:

- *Local Area Network (LAN)*, where, despite intense standards development, the VoIP equipment offered remains proprietary to a significant extent (e.g., in terms of integration of signalling functionality, compatibility, architecture, features and user interface design).
- *Metropolitan Area Network (MAN)*, where vendors partially implement signalling and traffic standards for their access devices (e.g., gateways), develop their own ones and/or restrict interoperability (e.g., compatibility with other manufacturer's products) for commercial purposes.
- *Wide Area Network (WAN)*, where providers deploy highly customised (and, to some extent, proprietary) solutions for IP-based voice transportation, with static configuration (e.g., fixed network topologies) being preferred due to the lack of scalable automated alternatives (e.g., call routing protocols).

As a result, a new architectural framework for VoIP, addressing the above issues according to the author's long working experience and personal vision of a global ISTN, has been developed and is presented, analysed and evaluated in this thesis.

1.3 Problem Definition and Scope

Network scalability is a complex, multi-faceted problem, for which the development of solutions can extend far beyond the limited spatial and temporal scope of a PhD thesis. This fact mandates the selection of a small, representative, well-defined set of parameters to investigate, and a clear description of the context in which these parameters are understood in the course of the research. For *Large-Scale IP Telephony*, in particular, the very concept of *scalability*, the area of *IP Telephony* itself, and their interrelationship, must first be defined.

1.3.1 IP Telephony in Context

Considerable confusion exists about the way IP is perceived as a voice carrier technology [253], [295]. Despite implications to the opposite, often found in the literature and in everyday usage of relative terms, *Voice over IP* (i.e., the technology for offering various voiceband services over IP networks) differs from *Voice over the Internet* (which is a special case of VoIP) and from *IP Telephony* (which can be seen as a wider concept, referring to IP-based multimedia on the whole, including voice and video telephony services). In general, usage scenarios, traffic profiles, performance characteristics and functional requirements may vary significantly among these three classes of voice applications and so treating them as one is prone to errors; furthermore, the terminology itself is evolving. An alternative definition of IP Telephony, for instance, referring to the replication of telephony services over an IP network, also exists (although it, too, can be considered a special case of VoIP), whereas transportation of the two most popular multimedia services over IP is being increasingly referred to as *VVoIP* (Voice and Video over IP) [67].

This thesis acknowledges the above issues and remains primarily focused on Voice over IP, where the term "voice" actually implies not only human speech but actually voiceband data (e.g., modem and fax tones), as explained in Chapter 2. However, because the main technologies addressed (e.g., signalling and media handling protocols) are common for all networked multimedia, the term "IP Telephony" is often used alongside (or as a service-level generalisation of) "VoIP", to indicate this joint applicability. In many cases, this wider scope is reflected in the analysis of related protocols, where the term "media" can be seen in the place of the narrower "voice".

Complications also exist with relation to the particular version (4 or 6) of the Internet Protocol employed, as IPv6 has significant differences to IPv4 and can only interact with it via special translator functions (protocol gateways). While the operational framework, as analysed in Chapter 3, is expected to remain the same, the VoIP protocols involved will have to be modified, at least to some extent, in order to support VoIPv6; therefore, generalising from VoIPv4 cannot be considered a straightforward process [269], and hence Version 4 will always be implied throughout this text in all references to the Internet Protocol, unless otherwise explicitly specified.

Furthermore, the concept of IP Telephony is perceived in this work as extending beyond packet switching, on the assumption that, as they grow towards global scale, IP Telephony networks will have to encompass (or, at the very least, interoperate with) parts or the whole of one or more fixed channel infrastructures, collectively referred to as Switched Circuit Networks (SCNs) [282]. Such networks are the analogue Plain Old Telephone Service (POTS), the Integrated Services Digital Network (ISDN), Intelligent Networks (INs) and various types of Public Land Mobile Networks (PLMNs, including current mobile phone and wireless infrastructures). A global SCN integrating POTS, ISDN and INs is the General Switched Telephone Network (GSTN), and that will often be used throughout this thesis to exemplify non-IP circuit switched voice transportation; however, most of the time, the research analysed is applicable to other types of SCNs (e.g., mobile networks), depending on the interworking function (gateway) employed, as explained in subsequent chapters.

Finally, it should be noted that there is a certain amount of differentiation between wired and wireless IP Telephony applications. Although many VoIP standards are common for both worlds (for instance, SIP is the preferred signalling protocol for next generation mobile networks), wireless networking presents challenges (e.g., in terms of reliability, interoperability and security) that merit special consideration, which extends far beyond the space and time allocation of this text. Therefore, this thesis focuses on wired voice communication, although much of the discussion applies, at least approximately, to wireless networks as well.

1.3.2 Top-level Network View

The top-level architecture of a communications network (voice, data, or other) is synthesised by three basic components (Figure 1.2) [363]: the *terminal*, which is the interface to the network services for the end user; the *switch*, which offers access to the core of the network; and the *backbone*, which interconnects otherwise local or isolated network "islands" that can be geographically concentrated or dispersed worldwide. The backbone constitutes the heart of the network, which is implemented via a number of dedicated, special-purpose, digital devices, that can process information very fast and switch it rapidly towards its intended destination, transparently to the user (hence the typical representation of the backbone as a cloud). Physical connectivity within the backbone is achieved via wired or wireless links, whereas logical connectivity is maintained via the routing mechanisms, which are responsible for dynamically disseminating topology information and thus allowing the switches to make decisions about the best possible paths that information should follow.

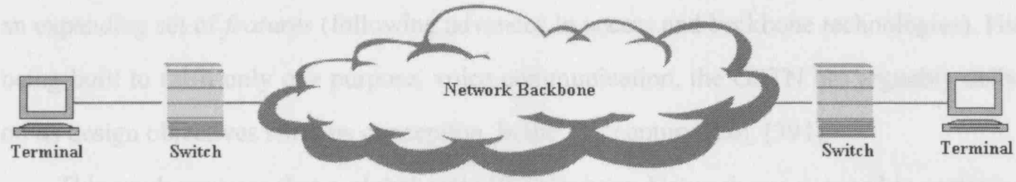


Figure 1.2: Top-level architecture of a communications network

This similarity in the top-level structure of voice and data networks, combined with the digital format used for transporting information of various types, facilitates the integration of communication services in a single infrastructure, which also complies with the same architecture, and IP Telephony networks are contemporary examples of such integration. It follows, according to the above, that, to discuss scalability in any network, all three areas affected by the aforementioned components, i.e. end user premises (LAN), access (MAN) and the backbone (WAN) must be examined, which is exactly the approach adopted by this thesis.

1.3.3 An Approach to Scalability

The success of a commercial communications network ultimately depends on the number of subscribers it can acquire and retain during its operation [9], [26], [223], [363]. Such a requirement, in turn, implies an unconstrained growth potential in the user base and, necessarily, in the technologies that implement it, so that the ever increasing traffic load is not translated to a noticeable service level degradation. This dual (user/network) growth potential is perceived as *scalability* in the context of this thesis.

For IP Telephony, in particular, as a non-disruptive technology, *user-level scalability* will depend on factors like ease of use, performance, QoS, security and pricing of the services offered, along the lines of established technology adoption paradigms [276], [309]; *network-*

level scalability, on the other hand, will be a synthesis of technical advances in the three areas already identified (LAN/MAN/WAN), and the effectiveness of these advances when implemented in Internet Protocol networks [253]. Both flavours of scalability are examined throughout this text and specific solutions are proposed, where appropriate.

1.3.4 Large-Scale IP Telephony

The General Switched Telephone Network is a worldwide SCN with a proven track record: *reliability* in the "five nines" ($\geq$ 99.999%) area, *scalability* (large user base and a global service accommodating it, in a distributed fashion, across the planet), sustained high voice *quality* and an expanding set of *features* (following advances in access and backbone technologies). Having being built to fulfil only one purpose, voice communication, the GSTN has arguably delivered on its design objectives since its conception, in the 19th century [26], [394].

This work assumes that a global scale IP Telephony Network can succeed in replacing the GSTN only by offering similar (or better) levels of performance: it will have to accommodate billions of voice terminals, interconnected through a wide variety of intermediate devices, wired or wireless, and grouped together into specific administrative domains, for efficiency, management and billing purposes; its providers will operate these domains under Service Level Agreements (SLAs) with other providers, as well as with end users (customers), at high levels of reliability and QoS; the carrier mechanism will be IP (version 4 [297] and, later on, version 6 [90]), enhanced by its companion Transport Layer protocols (TCP [298], UDP [296], or a newer one, such as SCTP [355]); network access technologies (layers 1 and 2 of the OSI Reference Model) will evolve according to the location, topology and policies of providers; Ethernet [263], Frame Relay (FR) [130] and Asynchronous Transfer Mode (ATM) [258] will remain popular candidates, but since IP can already co-operate with a wide variety of other protocols, it will keep the global network isolated of lower layer implementation details [63]. Furthermore, it is also reasonable to hypothesise that, even if IP Telephony succeeds in the performance objectives stated, it will also have to offer seamless interoperability with the ubiquitous GSTN for a significant transition period, in order to be socially acceptable and adopted by users [315].

For the purposes of a PhD thesis, the above assumptions, combined with the constant increase in networking hardware performance [363], allow the redefinition of the large-scale IP Telephony problem around the software/architectural aspect, and in particular around the application layer. This covers a significant part of both user-level scalability (through the design of user-friendly applications) and network-level scalability (by offering the performance and reliability levels required), and so it will be the main focus for innovation and further study for the rest of this work.

Concept	Description
ISTN	The global IP Switched Telephone Network
IPTN	Any IP Telephony Network, part of the ISTN
LIFT	The Large-Scale IP Telephony Framework, a new architecture for the ISTN

Table 1.1: Key concepts introduced by the thesis

1.3.5 Key Concepts

Within the context presented earlier on, this thesis introduces and makes consistent use of the following terms (Table 1.1):

- *IP Switched Telephone Network (ISTN).* The ISTN is a global voice network based on IP and offering universal connectivity to both circuit switched and packet switched terminals. It can be seen as the next step in the evolution of telephony services which, until a fully packet-based infrastructure is deployed end-to-end, will offer seamless interoperability with SCNs, via special devices (gateways).
- *IP Telephony Network (IPTN).* A possibly isolated part of the ISTN, which runs native IP Telephony protocols. An IPTN can be seen as corresponding to a subnet, as defined in the field of data communications.
- *Large-Scale IP Telephony Framework* (abbreviated as *LIFT* henceforth). LIFT defines the core architecture of the ISTN, including specifications for the LAN, the MAN and the WAN, in a different to current similar frameworks manner.

A number of additional terms used in the thesis describe LIFT-compliant implementations of ISTN components and are analysed in the corresponding chapters.

1.4 Hypothesis

This dissertation is based on the assumption that the aim of creating a universal IP Telephony infrastructure (the ISTN), which is sufficiently scalable to be deployed globally and also flexible enough to accommodate future developments without further redeployment, is highly desirable, as evidenced by current commercial and technological trends. Implementing such an infrastructure will inevitably require significant interventions in all three areas of end-to-end communication: the LAN (end users), the MAN (access networks) and the WAN (backbone).

It is the thesis of this dissertation that, in each of these areas, existing proposals are either partial or inadequate for supporting the desired aim, and that changes to core components of all three (specifically those to media clients, gateways and call routing) are an essential prerequisite to the achievement of that aim. Furthermore, such changes should be effected with an IP bias, but, crucially, without ignoring the benefits of the field-proven GSTN model.

1.5 Methodology

The stated hypothesis is supported by both qualitative and quantitative research methods, applied throughout the rest of this text.

1.5.1 Qualitative Strategy

In the investigation for a solution to the development of a large-scale IP Telephony network, this work follows a "funnel" approach: after an initial assessment of the generic *Voice over Packet Networks* area, and acknowledging the dominance of the Internet Protocol, it focuses on *Voice over IP* in order to develop a large-scale architectural framework that takes into account the advantages of the GSTN (particularly in universal connectivity, route stability and scalability), while simultaneously being applicable to IP Telephony.

The research presented herein is backed by an extensive literature review, which, in addition to a critique of performance and applicability for each of the various techniques employed, is detailed for all technologies (e.g., signalling and traffic protocols) widely used in this work and summarised for the remaining ones, thus making the thesis reasonably self-contained by design and providing the cognitive background necessary for the development of large-scale IP Telephony. The literature review is also used for revealing the inadequacies of existing solutions and for justifying the selection of specific standards (e.g., IP for packet switching, SIP for VoIP signalling and Java as the main programming language), whenever such a selection is available (e.g., RTP is the only choice for media encapsulation, so no further justification for its use in the applications developed, is due). Powerful development tools have been used for the implementation part, and the software written was tested with off-the-shelf equipment, thus allowing for a significant performance boost if state of the art hardware and advanced optimisation techniques are employed instead.

1.5.2 Quantitative Evaluation

The IP Telephony architecture presented in this thesis is summarised in the framework (LIFT) described and evaluated for *innovation* (differentiation) and *applicability* (implementation feasibility) in Chapter 4. For the empirical assessment of the architecture, one representative case from each of the three technological fields identified (the LAN, the MAN and the WAN) has been selected, as summarised in Table 1.2; more specifically, according to the problem definition and scope, the following application layer, software/architectural areas have been investigated in detail:

- The *User Agent* (i.e., the client application), representative of the LAN field and necessary for interfacing the end user with the IP Telephony network. The improvements contributed in this area are analysed in Chapter 5 and evaluated for implementation complexity and operational performance (CODEC characteristics).
- The *Gateway*, representative of the MAN field and, among other potential uses, offering interoperability among the various circuit-switched and packet-switched "islands" of the IP Telephony network. The improvements contributed in this area are presented in Chapter 6 and evaluated for signalling (simultaneous call support) and media (comparative media engine) performance.
- The *Call Routing* mechanisms (topology structure and associated protocols), representative of the WAN field (i.e., the backbone) and maintaining connectivity inside and across administrative domains of the IP Telephony network. The improvements contributed in this area are analysed in Chapter 7 and evaluated for performance (convergence time) and scalability (routing space size).

Communication Area	Representative Application
Local Area Network (LAN)	User Agent
Metropolitan Area Network (MAN)	Gateway
Wide Area Network (WAN)	Call Routing

Table 1.2: End-to-end connectivity areas and their representative applications

The evaluation of the above three implementations provides results that allow a generalisation to the entire problem, on the reasonable assumption that a solution which can be completed within the few man years available to a PhD, can easily appear as a robust, scalable product in a much shorter time frame, if a proper amount of resources is committed to such a purpose.

1.6 Contributions

This thesis introduces a number of contributions, the most important of which can be summarised as follows:

- A new framework for large-scale IP Telephony networks, *LIFT*, based on the "call anyone, anywhere, anytime" model of the GSTN, but with an IP bias (Chapter 4).
- A new IP Telephony *User Agent* architecture, which improves existing designs in terms of completeness, functionality, configurability and usability. Applications compliant with this architecture, such as the *CAT* presented in Chapter 5, are capable of offering GSTN-level services to the VoIP user.
- A new VoIP *Gateway* architecture, which modifies the current perceived standard, for scalability and modularity purposes. The proposed architecture allows transparent interoperability between the GSTN and any VoIP network, independent of usage scenarios, thus facilitating the establishment of a fully integrated ISTN; it has been implemented in the *VIA*, analysed in Chapter 6.
- A new *Call Routing* mechanism, *HIT*, which augments the current, nearly flat topological model of the IP Telephony universe of terminals, switches and other devices, in favour of an improved, more hierarchical one, thus accommodating much higher host numbers; an elaborate estimation of ISTN growth perspectives is also conducted and an implementation of HIT, *HITRA*, is evaluated in Chapter 7.

Secondary contributions appearing in this thesis include a number of earlier achievements related to packet voice (some of which have been materialised into products, either research or commercial), the extension to the UCL-CS Robust Audio Tool (RAT) for ATM support, an analysis of TRIP (the current protocol for IP Telephony call routing), and a set of ideas for further research related to IP Telephony, particularly in the areas where this work has focused.

The above contributions are described in detail in a number of related technical reports and conference publications, listed in the Bibliography section. A number of additional publications, based on the findings of Chapters 2-7, are also foreseen.

1.7 Thesis Organisation

This thesis is organised so that the included material progressively supports the stated hypothesis. More specifically, the focus of each of the eight Chapters is:

- *CHAPTER 1 (Introduction)*: Comprehensive summary of the research problem, the motivation to develop a solution, core hypothesis, methodology and an overview of the actual work to be presented in the rest of the document.
- *CHAPTER 2 (Voice over Packet Networks)*: A survey of packet-switched voice, identification of its advantages over circuit-switched voice, analysis of the common implementation techniques still applicable today and justification of why IP is the preferable solution for realising a large-scale packet voice network.
- *CHAPTER 3 (IP Telephony)*: Detailed overview of the main technologies related to the selected packet voice area (VoIP), including an analysis of the 3 basic components dealt with in the thesis: the User Agent, the Gateway and Call Routing.
- *CHAPTER 4 (A Framework for Large-scale IP Telephony)*: Description of LIFT, a generic architecture for large-scale IP Telephony Networks and evaluation of its characteristics, compared to other similar specifications.
- *CHAPTER 5 (An IP Telephony User Agent)*: Design and implementation of CAT, i.e., the first fundamental component of the ISTN, the IP Telephony User Agent; differences with existing alternatives; evaluation of the developed audio tool.
- *CHAPTER 6 (An IP Telephony Gateway)*: Design and implementation of the VIA, i.e., the second fundamental component of the ISTN, the IP Telephony Gateway; differences with existing alternatives; evaluation of the developed gateway.
- *CHAPTER 7 (An IP Telephony Call Routing Protocol)*: Design and implementation of a Hierarchical IP Telephony Call Routing Protocol (HIT), as an extension to the current, BGP4-based TRIP, accompanied by an extensive analysis of ISTN growth prospects; differences of HIT to existing alternatives; implementation and evaluation of the developed call routing protocol.

- *CHAPTER 8 (Conclusions and Future Research)*: Suggestions for expansion of this work, including a number of ideas of current research interest in the IP Telephony area.

An *Overview* section at the start of each Chapter summarises its role in the thesis and lists the main findings that follow. Chapter 1 sets the exact scope of this research, whereas Chapters 2 and 3, complying with the funnel approach and the self-contained philosophy of the thesis, are dedicated to an extensive literature review, combined with a description of the standards examined or implemented later on. For Chapters 4-7, which constitute the research part, the overview section is followed by another one that provides an in-depth critique of closely *Related Work*, thus specialising the discussion found in Chapters 2-3 and revealing the inadequacy of current solutions for realising the ISTN. An *Evaluation* section at the end of Chapters 4-7 analyses and discusses the corresponding research results, while a *Conclusion* section at the end of each of the 8 chapters summarises its main findings. Finally, an extensive set of *References*, separated in *Bibliography* and *Online Resources*, is used, and a number of *Appendices* are included at the end of the document.

1.8 Conclusion

IP Telephony represents a wide area of applied and potential research, with a large number of issues remaining unresolved and an intense development effort currently ongoing, due to the benefits associated with the technology and the huge interest thus expressed for it by both the academic and the commercial world.

This thesis explores a specific such unresolved issue, *scalability*, which it addresses, within a well-defined scope, by investigating three core communication areas: end user premises (LAN), network access (MAN) and the backbone (WAN). A single representative case from each of these areas (the VoIP user agent, gateway and call routing mechanism, respectively) has been selected for qualitative and quantitative analysis, and novel solutions are presented in the following chapters. This way, the applicability of large-scale IP Telephony networks is shown on a step-by-step basis, as per the stated hypothesis, and an integrated architecture for their implementation is subsequently introduced.

CHAPTER 2

Voice over Packet Networks

2.1 Overview

The use of data networks for voice transportation dates back to the origins of packet switching, in the early 1970s. Progressively, this practice led to exploiting the technology for the creation of integrated services networks, in which voice, data, and other, more resource-demanding media types (e.g., wideband audio and video) are supported, thus paving the way for a multi-service communications infrastructure, alternative to circuit switching networks like the GSTN.

This chapter provides a generic review of packet voice, analysing the main techniques for its implementation and common workarounds for the problems associated with it, in a network and product-agnostic way, complying with the self-dependent structure introduced in Chapter 1. More specifically, no particular transport technology is assumed and those principles are discussed that are applicable to most packet voice protocols, including IP, Frame Relay and ATM. In the end, a comparison of these three standards reveals the superiority, in practice, of IP, thus justifying the selection and closer investigation of VoIP, which takes place in Chapter 3 and onwards. Many of the techniques presented in the following sections have been continuously implemented for over a decade in the majority of IP Telephony solutions and products, so they are incorporated as such in the work analysed in Chapters 4, 5, 6 and 7.

2.2 A Brief History of Packet Voice

Circuit-switched telephony started moving away from its late 19th century analogue operational model in the early 1960s, with the deployment of the first digital circuits in the United States [26]. This evolution, combined with the development of (natively digital) packet-switched data networks near the end of the same decade [363], led to the first attempts to integrate these two different types of information over a common backbone, in the early 1970s.

Successful experiments for packetised voice transportation have been recorded from as early as 1971 at MIT and 1972 over the ARPANET, producing working solutions of point-to-point or conferencing connections in a variety of cases (wired ARPANET, packet radio PRNET, satellite SATNET) [127], [132], either proprietary [389], or more standardised, after the release of the Network Voice Protocol (NVP) in 1977 [76]. These efforts exploited long experience on circuit-switched voice and, in particular, advances in the fields of encoding and digital transmission that started during the 1930s, were field-proven during World War II and progressed rapidly during the 1950s and 1960s, leading to the first publicly available digital telephony services [26], [127]. Still in the 1970s, an early version of ST (the Internet Stream Protocol [369], a connection-oriented alternative to IP [297]), emerged as the most popular choice for supporting voice communication with rudimentary resource reservation, guaranteed Quality of Service (QoS) parameters and interoperability with circuit-switched voice, achievements arguably impressive for the time. This way, the feasibility and practicality of packet voice, at least as a complementary service to conventional telephony, was verified [127], using what could be called the "first generation" of audio tools (Table 2.1).

Generation	Period	Example
First	1970s-1980s	Experimental applications
Second	1990s	vt, vat, NeVoT, RAT, FreePhone
Third	2000 and on	Skype, Gizmo, GoogleTalk, X-Lite

Table 2.1: Audio tool generations

Activity in the area intensified during the 1980s [31], [138], [389] and early 1990s with preference gradually being switched to IP [7]. In 1991, the NVP-based Voice Terminal (vt) from ISI [U16] was used for conducting the first packet audio conference, over DARTnet [257]. Later on that year, the successor of *vt*, the Visual Audio Tool (vat) [U58] from LNBL [U26], started being used over the MBONE [257], initially in the United States [286]. In March 1992, VoIP crossed over to Europe and Australia, when the first IETF Internet "audiocast" (i.e., packet audio transmission of the then IETF meeting in San Diego) interconnected 20 sites across three continents (America, Europe and Australia) via the Multicast Backbone (the MBONE) [103], [252] with UCL-CS [U56] joining from the United Kingdom [60]. This development represented another milestone in the long history of packet voice communications, as *vat* significantly influenced the development of numerous similar applications by the academic world that appeared in the same period, including the Network Voice Terminal (NeVoT) at the University of Massachusetts [330], the Robust Audio Tool (RAT) at UCL-CS [153], [154], [155], [163], [232] and the FreePhone at INRIA [37], [38] (the latter two partially being products of collaboration under the MICE project [329]).

The VoIP revolution eventually reached the wider public in 1995, with the release of the free "Internet Phone" application by VocalTec [70], [U59], which was soon followed by several other shareware tools, such as Speak Freely [U49]. Also in 1995, the first streaming software, RealAudio [U41], was released by Progressive Networks [299]. At the same period (particularly during 1992-1994), a number of applications for video over IP were developed, starting with CU-SeeMe from Cornell University [96]; the Network Video (nv) tool from XEROX PARC [117] and the INRIA Videoconferencing System (IVS) [372] that followed influenced the development of the MBONE videoconferencing tool, *vic* [256]; all these utilities integrated (or cooperated with) VoIP applications (e.g., *vat* or *RAT*). Microsoft, too, contributed to the popularisation of audio/video applications with the development, in 1996, of NetMeeting [U29], which was bundled in subsequent versions of Windows. In 1996, Net2Phone [237], [U32] pioneered the concept of IP Telephony Service Providers (ITSPs) by releasing software that allowed calls to the GSTN to selected destinations in the United States and abroad, where the company had gateways installed.

Audio and video conferencing over IP was conducted using proprietary mechanisms for call setup until the mid-1990s, although two similarly named tools, the session directory (sd) from LNBL [U26] and the session directory (sdr) from UCL-CS [U56], became a quasi-standard for the MBONE [252]. In 1996, the three major IP Telephony standards, H.323 [188], SIP [318] and RTP [337], appeared in standardised form, at different levels of maturity [286]. Soon, in the late 1990s, not only the suitability, but also the dominance of IP over other packet technologies for large scale voice transportation was confirmed, particularly after the failure of ATM to reach the desktop, despite its clear technical advantages [250], [258], [393], [395]. This can be considered as the concluding era of the "second generation" of audio tools.

The "third generation" begun at the start of the 21st century, when packet voice, in the form of IP Telephony, had already evolved into a mainstream technology for both terrestrial [128] and space communications [358]. The products of this generation are closer (or identical) to the "softphone" model discussed in Chapter 3, which, in addition to ongoing research for more advanced characteristics, is increasingly geared towards commercial standards, incorporating sophisticated Graphical User Interfaces (GUIs), often a telephone-like look and feel (e.g., Skype [U48], Gizmo [U12], GoogleTalk [U13]) and other complex features like security (e.g., PacPhone [U38]). Most IP Telephony equipment vendors provide a customised voice client compatible with their product line, resulting in several different (and not always compatible) offerings [U22]. A common characteristic of many of these tools is the tighter integration with other services beyond voice, such as video, chat, file transfer and instant messaging.

In retrospection, it can be argued that the research conducted in the area until the mid-eighties has led to discoveries which remain applicable, with the added benefit of being considerably network-agnostic, i.e., without any bias towards IP, as can be seen in the rest of this chapter. Furthermore, the evolution of packet switching technology has made possible its use for a variety of traffic types beyond voice, in a stand-alone or an integrated fashion. However, transportation of voice (more generally, voiceband data) remains the least demanding, easiest to implement and, thus, most popular multimedia service over a packet network.

2.3 Voice Transportation

Voice belongs to a class of media types that require timing constraints for their successful transmission over a communications network, thereby generally referred to as real-time [7], [245]. These media types and the applications handling them can be broadly classified as *soft real-time*, which exhibit some tolerance to delay and packet loss, and *hard real-time*, which do not. Voice and video are the most frequently encountered soft real-time media.

Audio Category	**Spectrum (Hz)**
Telephony ("voiceband")	300-3,400
AM radio ("wideband")	50-7,000
FM radio	50-15,000
Compact Disk (CD)	20-20,000
Human audible	20-20,000

Table 2.2: Typical spectra of various audio types

2.3.1 Characteristics of Voice

Signal processing theory [113], [285] has shown that frequencies below 3,500 Hz represent the most semantically significant portion of human speech, hence leading to the 4 KHz (0 Hz to 4,000 Hz) "voiceband" spectrum being used for the telephony network, ever since it was invented in the 19th century [26]. Signals falling in the above frequency range are collectively referred to as "voice" and can be anything from conventional data, fax, or digital tones (e.g., modem or DTMF), to human speech itself. Attempting to transmit higher frequency audio types (Table 2.2) [26], [211], [245] over the telephony network results, unsurprisingly, in quality degradation, although such attempts have been recorded [138]; these audio types are, however, supported in higher quality packet voice applications [394].

Generally speaking, the comparison between voice and data represents a tradeoff between the needs for timing and accuracy. Seen from a unified point of view, voice can be handled as a special class of data, originally produced by a constant bit rate source. Like data, it is transportable in digital format and can be described as bursty and intermittent, which allows significant bandwidth efficiencies, when combined with the statistical multiplexing nature of packet networks. On the other hand, it is much more time sensitive, but also more loss-tolerant compared to non-real-time data, hence requiring significant enhancements to the conventional packet switching service; these enhancements, depending on technology, are not always feasible (for example, X.25 networks could not normally achieve the delay targets imposed by a voice stream [176], [363], whereas their successors, based on Frame Relay, generally can [130]). Furthermore, each of the various voice signal types is characterised by its own, unique set of properties, which must be taken into account for its transmission over a packet network.

Speech, although of relatively low data content ("speech as text" would only require around 50-70 bits/sec [113], [127]), carries additional information, such as emotion, pitch and gain, which can raise bandwidth requirements by 3 orders of magnitude (up to several tens of Kbps) [114], [348]. However, speaker pauses during monologues, long silent periods within conversations, repetitive signal patterns in voiced sounds, easily parameterised unvoiced sounds and other redundancies in the time or frequency domains, combined with the masking (filtering) properties of the human ear, can reduce overall capacity requirements to well below 50% of their nominal values [26], [73], [113], [229]. Fax traffic, on the other hand, cannot be "compressed" to that extent, as it is used for inter-machine communication; in addition, it is less error-tolerant and more delay sensitive than speech, because of its data nature and the associated protocol timeouts. Similar characteristics are demonstrated by other voiceband data types, like modem and DTMF tones.

Unless otherwise specified, the term "voice" is used in this work to refer primarily to real-time, interactive speech, although most of the discussion applies to the other types of voiceband data, as well.

2.3.2 Advantages and Challenges of Packet Voice

A number of compelling reasons have propelled the replacement of analogue voice networks with digital ones in the telephony world during the past decades [26]: signalling was greatly simplified; robustness to channel errors significantly improved; multiplexing became much more efficient; integration of transmission and switching equipment has been made possible; and the intelligence of the network has been vastly increased, due to the introduction of computers. These advantages have allowed operators to offer lower cost, better quality and more feature-rich services, thus well justifying the universal adoption of digital voice in contemporary telephony networks.

The general advantages of digital technology apply not only to the “analogue vs. digital telephony” question, but also to the “circuit vs. packet switched communications” dichotomy. Packet switching networks use Statistical Multiplexing (SM) and Bandwidth-on-Demand (BoD), compared to their circuit switching counterparts, which rely on less resource-efficient techniques, like Time Division Multiplexing (TDM, as in landline telephony) and Frequency Division Multiplexing (FDM, used in mobile networks) for implementing their voice transportation service. Packet networks are therefore better when it comes to accommodating bursty voice traffic profiles.

Packetised voice communication offers a number of additional benefits, including simplified administration procedures, training and network management; significant bandwidth economies through compression and silence suppression; overall cost savings; a unified network architecture for integrating different types of media; distributed/component-based functionality; rich participant information; reduced call blocking probability; enhanced security features such as encryption; intelligent terminals; and advanced network services, like instant messaging, user presence and multimedia conferencing.

The main challenges associated with packet switching are related to its performance with respect to real-time multimedia traffic and, particularly, the difficulty to offer the required Quality of Service (QoS) levels. In fact, replicating the performance, reliability, QoS and functionality of the GSTN remains a major objective for packet voice networks, which has proved difficult to achieve fully. Earlier efforts to address the issue by using circuit switching principles, resulted in the invention of techniques like message switching (fixed-path transmission of packets) [373] and hybrid switching (TDM-based packet transmission) [137], but implementation complexity, combined with non-ideal results, maintained native packet switching as the voice transportation technology of choice for contemporary networks.

From a more theoretical point of view, packet-switched voice represents a step before the conclusion of a fundamental evolution cycle: analogue voice, digital voice, digital data, digital voice and data, integrated multimedia networks.

2.3.3 Interoperability

The perceived superiority of packet switching does not change the fact that there are universal SCNs, like the GSTN and the various mobile networks, which successfully accommodate voice communication needs worldwide. Since replacing these SCNs is not feasible (or reasonable) in terms of investment and effort, for packet voice to become global it is necessary that full interoperability ("interworking") between packet networks and their circuit switching counterparts be achieved. This is not, however, always a straightforward task and, indeed, the difficulties associated with it can be seen as one of the reasons why packet voice has not yet reached the desktop in a universal scale.

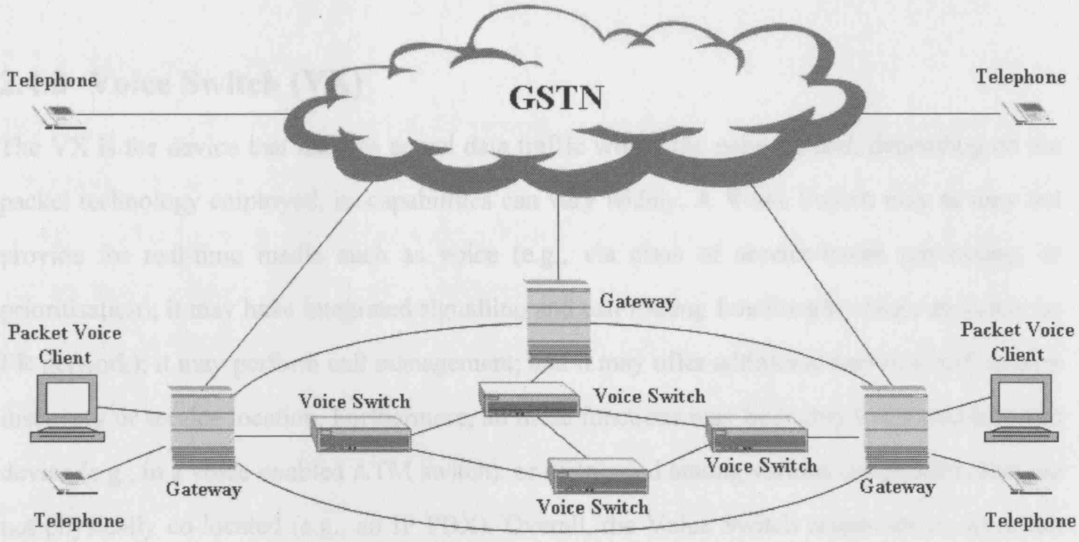


Figure 2.1: High-level architecture of a packet voice network

2.4 Architecture of a Packet Voice Network

The top-level architecture of a voice-enabled packet switched network resembles closely that of the conventional telephony backbone. Like the GSTN, service operation is partitioned into several different logical functions, which are implemented in physical devices distributed across the entire network. As shown in Figure 2.1, the main logical components of such an (abstracted) architecture are the *Packet Voice Client (PVC)*, the *Gateway (GW)* and the *Voice Switch (VX)*. All other entities encountered in a packet voice network can be treated as functional variations of the PVC, the GW and the VX, as analysed in the next sections.

2.4.1 Packet Voice Client (PVC)

The PVC is a device running application software that enables it to act as either a Packet Voice Transmitter (PVT), or a Packet Voice Receiver (PVR), under user instructions. Typical examples of such devices are:

- An application ("audio tool") with integrated or distributed signalling and media capabilities.
- An intelligent stand-alone voice terminal (e.g., an ATM phone).
- A legacy device connected to the network via a special adapter (e.g., a conventional telephone accessing the IP network via a PC with a voice card, or via an intelligent PBX).

The PVC is the packet-based entity corresponding to a GSTN telephone.

2.4.2 Voice Switch (VX)

The VX is the device that handles actual data traffic within the network and, depending on the packet technology employed, its capabilities can vary widely. A Voice Switch may or may not provide for real-time media such as voice (e.g., via class of service-based processing, or prioritisation); it may have integrated signalling and call routing functionality (e.g., as within an FR network); it may perform call management; and it may offer additional services such as user discovery or service location. Furthermore, all these functions may be highly integrated into one device (e.g., in a voice-enabled ATM switch), or distributed among various components that are not physically co-located (e.g., an IP PBX). Overall, the Voice Switch resembles in operation the various types of switches found at the core of the GSTN [26] and, indeed, it is the basic building block of the packet voice network backbone.

2.4.3 Gateway (GW)

Gateways are devices that secure interoperability between Switched Circuit Networks like the GSTN or mobile networks [282], and Packet Switched Data Networks (PSDN), by converting ("translating") between SCN and PSDN voice signalling and traffic formats, in a transparent to the end user manner. More specifically, gateway functionality typically includes user-level call management, SCN and PSDN call termination, as well as media format translation. This functionality can be provided via a fully integrated device, or a distributed network of components, communicating internally via special mechanisms such as the MGCP [5] and

H.248/MEGACO [134], [187] protocols used in Voice over IP [128], as discussed in Chapter 3. Importantly, gateways operate as endpoints in each of the different networks they interconnect, that is, they terminate every incoming call and re-initiate it for the other side of the connection, after they perform the necessary adaptations (e.g., protocol conversion) [35], [77], [133], [315].

Implementations of a gateway may vary from conventional PBXs that have been enhanced with packet capabilities, to full-blown voice-enabled packet switches (e.g., routers with voice expansion cards). In many cases, additionally to the typical packet voice interface, gateways offer analogue (2/4-wire) and digital (T1/E1) ports, for connectivity of conventional GSTN equipment. From a functional point of view, a gateway is very close to a Private Branch Exchange, although, in a packet voice network, PBX operations tend to be distributed across more than one device.

2.4.4 Other Entities

The high-level picture of a packet voice network presented in Figure 2.1 and analysed above corresponds to Layers 3-7 in the OSI Reference Model, and is not, of course, complete. A wide variety of other functions implement the end-to-end service, either in a stand-alone fashion (e.g., autonomous hardware components), or as processes co-located in larger physical devices. Depending on the packet technology employed each time, a typical configuration may also include LAN access devices (e.g., Ethernet switches [363]), directory services (e.g., LDAP [384] or DNS [273]), call routing functions (e.g., TRIP location servers [314]), signalling servers (e.g., an H.323 gatekeeper [188]), higher-level switches (e.g., a PNNI aggregated node [11]), or specialised traffic processors (e.g., a voice transcoder or a conferencing mixer). For the presentation and analysis of the core issues related to packet voice transportation, however, the key components are the Packet Voice Client, the Gateway, the Voice Switch (mainly from a QoS and call routing point of view) and the communication mechanisms these devices implement, understand and comply to, as already discussed.

2.5 Call Management

The typical procedure for establishing a voice connection ("call") among two or more communication endpoints is similar for both circuit and packet switched networks, and can be summarised into three discrete phases:

- *Call Setup.* A sequence of signalling messages originated by the calling terminal, which enable it to interact with the network and eventually create the desired voice connection. After acceptance of the setup request by the called terminal, the network puts in place other call management mechanisms (such as, for instance, accounting and billing) that operate throughout the rest of the call.
- *Information Transfer.* Actual transfer of the voice data (e.g., human speech captured during a conversation, or a number of pages sent in a fax call) among the communicating endpoints.
- *Call Release.* The tearing down of the connection, caused by any of the communicating parties or, in rarer cases, the network itself (for example, because of overload, or a link failure).

The above procedures take place simultaneously in two different levels: the *Control Plane*, where signalling entities operate, and the *User Plane*, where traffic-related functions are executed. Both require significant enhancements to existing data mechanisms, in order to efficiently accommodate voice. These enhancements range from simple feature support (e.g., capability negotiation) to techniques for achieving the minimum performance targets acceptable for a real-time voice service, as understood and implemented in conventional telephony networks [26].

2.5.1 Control Plane (Signalling)

The control plane does not impose so strict timing requirements as the user plane, thus, from a performance point of view, targets relating to signalling are the easier to meet. However, to accommodate voice, for most packet network technologies, improved and/or new signalling mechanisms are often needed.

Particular challenges for packet-based voice signalling protocols include:

- **Support of the GSTN feature set:** At least the most popular features - like user alert, busy, redirection, call forwarding and location dependence (e.g., country-specific) or independence (e.g., 0800 numbers) - must be offered, while Intelligent Network (IN) features [120], [203] are also highly desirable.
- **Interoperability with SCNs:** For packet voice to become attractive to users at a large scale, seamless integration with circuit-switched networks such as the GSTN and the various mobile telephony networks must be achieved.
- **Session management:** Connectionless services (like IP) essentially simulate connection-

oriented operation at the application layer, by maintaining communication associations ("sessions") among call participants. This is necessary not only for achieving the desired QoS, but also for procedures like admission control and capability negotiation, usually found only in connection-oriented technologies, like ATM. Even there, however, signalling mechanisms need to be augmented (e.g., by including voice-specific parameter negotiation).

- **Call routing:** Call routing for voice communications may be call-oriented (as in the GSTN), packet-oriented (typically in IP Telephony), or both. Depending on the network architecture, the mechanism selected directly affects the traffic path, as well, particularly for connection-oriented protocols, such as ATM.

  Because voice traffic has specific service requirements, often the routing mechanisms used for conventional data are not enough, and QoS-based schemes are needed. Depending on technology, this may translate into complex overlay routing mechanisms (for example, TRIP [314] for IP, or PNNI [11] for ATM). This complexity increases in hybrid environments, where the network nodes must be intelligent enough to switch voice between the packet and the circuit switched subnets, depending on parameters like cost, availability, quality, or even user location. Furthermore, in packet voice networks, call blocking becomes a function of the average bandwidth required (as opposed to peak bandwidth in circuit switching) and recovery ("crankback") is complicated by the per-packet (as opposed to per-call) processing principle, particularly in the case of connectionless architectures.

- **Billing:** There is no general consensus as to how charging for a packet voice service should be conducted [82]. Connectionless packet services like IP, typically maintain little or no state at all, thus complicating things even further. Billing may prove essential for attracting the amount of investment necessary for the universal deployment of packet voice services.

Overall, signalling operations can be implemented in software, hardware, or a mixture of both. Connectionless services (like IP) favour the first approach (e.g., SIP [318]), while connection-oriented ones follow the second (e.g., ATM switches [258]).

2.5.2 User Plane (Traffic)

The user plane is characterised by the (soft-to-hard) real-time nature of traffic (including voice), thus mandating maximum efficiency in the numerous operations involved. These operations can generally be implemented in software (e.g., RTP), hardware (e.g., DSPs), or a mixture of both (e.g., an ATM phone); they are all analysed next, in an order roughly corresponding to the journey of a packet from source to destination.

2.5.2.1 Recording

The original voice signal is captured, in analogue form, from the environment, using appropriate hardware (e.g., a microphone in the case of speech), to be processed and transmitted to the destination. Variations in the fidelity of the capture equipment can affect the quality of the transmitted voice as perceived by the receiver.

2.5.2.2 Encoding

The voice signal is converted from its natural, analogue form, to digital, and then formatted appropriately, prior to transmission.

Encoding is a process implemented in two stages, sampling and quantisation [348]. *Sampling* reduces the infinite number of amplitude levels existent within any time interval, to a finite set of values. *Quantisation*, i.e. "adjusting" those values to binary numbers, is unavoidably an approximation process, which, depending on the encoding method employed, may result in non-negligible loss of quality (referred to as *quantisation noise*). After the digitisation of the original (analogue) voice signal, techniques for saving ("compressing") bandwidth are applied, depending on service specifications (e.g., desired user quality). Appropriate CODEC (COder/DECoder) devices are used for this purpose, which are capable of reversing the procedure at the receiver. In fact, apart from digitisation and compression, certain CODECs (e.g., the ITU-T G.729 [183]) are also capable of packetising and de-packetising voice directly, thereby increasing efficiency.

Encoding as a technology has been extensively researched since 1928, when the "channel vocoder" appeared [26], and in the following decades [55], [113], [127], [348]. Depending on the particular digitisation/compression methodology followed, there are three main types of CODECs [73], [114], [229]: *Waveform* CODECs attempt to faithfully represent the original speech waveform; *Predictive* CODECs encode only the perceptually significant aspects of speech; and *Hybrid* CODECs use combinations of these two methods. Non-speech voice signals, such as modem or fax tones, are usually tolerant to waveform compression only [229].

The encoding process imposes a non-negligible delay penalty, since a number of samples (the *audio unit*) need to be collected before the digitised form of the original voice stream can be compressed. In general, there is always an unavoidable trade-off: increased compression results in increased delay, increased processing load and decreased voice quality [114], [348].

2.5.2.3 Silence Suppression

One of the main advantages of packet voice systems is their ability to exploit periods of inactivity ("silence") to reduce bandwidth consumption. This is a two-step procedure: First, special algorithms that monitor changes in the power and frequency of the voice signal identify active intervals of a voice source - a process called Speech Activity Detection (SAD), or, more generally, Voice Activity Detection (VAD); then, during silence, output from other traffic sources is multiplexed in the network - a process called Digital Speech Interpolation (DSI) and representing the evolution from the corresponding, rather complex analogue method, Time Assignment Speech Interpolation (TASI) [26] used in some circuit switched systems.

In interactive speech communications, in particular, the speaker pauses between syllables, words and phrases, or to listen to the other speaker(s), which can yield inactivity periods of 60% or more [26], [113], [285]. This fact, combined with the findings of several studies regarding average speech activity and inactivity periods ([44], [154], [240], summarised in Table 2.3), leads to substantial economies of bandwidth. Silence suppression capabilities are built-in into certain CODECs, such as the ITU-T G.723.1 [182] and G.729 [183], whereas in others it is performed by separate devices, in tandem, prior to packetisation. Furthermore, in order to maintain the sense of an active line during "silent" periods, it is necessary that the sender periodically transmits (or the receiver locally generates) artificial background sounds ("comfort noise"). To facilitate this, CODECs can generate special Silence Insertion Descriptors (SIDs), which indicate the amplitude of the background noise to the receiver [73], [229]. Transmission of comfort noise packets happens at a much lower rate than normal voice traffic; hence, bandwidth savings are not significantly affected.

User	Network
Talkspurt duration	0.5 sec – 1.5 sec
Silence duration	1.5 sec – 2.5 sec
Packet duration	20 ms, 40 ms, 60 ms, 80 ms, 120 ms
Playout buffer size	40 ms – 200 ms

Table 2.3: Indicative time domain values for packet voice

It should be noted that silence suppression is not always a perfect process; because of inaccuracies in identifying the boundaries of active/inactive periods, background noise and other difficulties such as speaker characteristics, numerous side-effects, like front-end clipping, mid-speech clipping and hold-over may occur, which can affect the quality of the transmitted voice signal [138].

Encoding	**Type**	**Standard**	**Rate**	**Delay**	**MOS**
PCM	Waveform	G.711	64 Kbps	0.125 ms	4.30
ADPCM	Waveform	G.726	32 Kbps	0.125 ms	4.10
SB-ADPCM	Waveform	G.722	64 Kbps	3 ms	4.30
LD-CELP	Hybrid	G.728	16 Kbps	0.625 ms	4.00
CS-ACELP	Hybrid	G.729	8 Kbps	15 ms	3.95
MP-MLQ	Hybrid	G.723.1	6.3 Kbps	37.5 ms	3.90
CS-ACELP	Hybrid	G.723.1	5.3 Kbps	37.5 ms	3.50
MR-ACELP (AMR)	Hybrid	GSM 06.90	4.75-12.2 Kbps	20 ms, 25 ms	3.50-4.10
RTE-LTP	Hybrid	GSM 06.60	13 Kbps	20 ms	3.5
LPC	Vocoder	LPC-10	2.4 Kbps	22.5 ms	2.5

Table 2.4: Popular encoding standards

2.5.2.4 Packetisation

Encoded voice samples are assembled into packet payloads and prepared for transmission through the network. These samples typically correspond to a few tens of milliseconds ("packet duration") in the time domain [389], because of performance requirements and additionally, in the case of speech, due to basic phonetic properties of the original signal [113]. This requirement also stems from research for the optimal packet length for voice communications, which generally indicates packet durations of less than 50 ms and actual payload sizes of a few hundred bits [268], corresponding to tens of milliseconds in the time domain, given the rates of most CODECs (Table 2.4). Larger payloads could be used to improve bandwidth efficiency, but there is a tradeoff with timing, as bigger packets cause delays over slower paths, increase loss burstiness and reduce the effectiveness of packet loss recovery methods [210], [288]. In general, voice packet sizes (in bytes) are defined according to the desired packet duration, which is CODEC-independent, with most common values being 20 ms, 40 ms or 60 ms and, in rarer cases, 80 ms or 120 ms, as seen in Table 2.3.

Given the relatively small size of the payload, successive encapsulation of the voice samples, from the Application Layer down to the Data Link Layer, can lead to large overheads. Furthermore, to support the service requirements of voice applications, the conventional data link or network layer functionality has to be enhanced with special control information. The problem is more evident in connectionless networks that use self-dependent packets with full source and destination addresses; there, extra header fields, such as a sequence number (to detect lost packets) and a timestamp (to record timing) are necessary, further increasing overhead [286]. Various header compression schemes have been proposed as a solution [230], but are of limited applicability, due to their proprietary and lossy nature.

2.5.2.5 Serialisation

Voice packets are placed on the network interface for transmission to their destination. This *insertion* process is usually very fast, as it mainly depends on the packet size and the access link capacity, but it can be slowed down because of channel multiplexing and queuing at the sender.

2.5.2.6 Transportation

In addition to the physical bit transmission over the communications channel, transportation also includes switching inside network nodes, which may result in queue-related non-deterministic behaviour (e.g., highly variable delays), depending on the technology (connectionless or connection-oriented) and the topology (LAN/MAN/WAN) of the network. Interworking operations, such as signalling translation and media transcoding, may also be necessary, if the boundaries of the packet network are to be crossed towards an SCN.

Of particular importance in the transportation process is the selection of the voice packet size. Small, fixed-size packets are easier to accommodate in the event of delay or loss, but can be bottlenecked behind larger ones carrying data if special precautions (e.g., prioritisation inside network nodes) are not taken. Certain technologies (like IP) allow fragmentation for alleviating this effect, but this method can also be problematic due to processing overheads and a degradation of performance when one or more fragments are lost [221].

2.5.2.7 Deserialisation

After the network delivers voice packets to the receiver, the latter reads ("deserialises") them out of the channel and passes them to the application layer, for further processing.

2.5.2.8 Decoding

The received voice must be decoded, so that it can then be converted to its original, analogue form, and reproduced through the appropriate device. Decoding is one of the stages of the (complicated due to the packet-based communication) playback procedure.

2.5.2.9 Playback

The received media stream must be reconstructed and played back in its original (analogue) form, through appropriate hardware (e.g., a pair of speakers). The success of this process depends heavily on synchronisation, i.e., accurate reproduction of the timing interdependencies present during packet generation at the sender.

Statistical multiplexing causes fluctuations to the latency that packets experience in their journey through the network. This delay variation (i.e., the difference between maximum and minimum delay experienced by packets in a session, also called "jitter") is compensated for, at the receiver, via a dedicated queue, the playout buffer, which is also used to correct other side effects, such as lost, out-of-sequence or duplicated packets in connectionless networks. (Other timing and frequency fluctuations, including "swim", "wander" and "drift" [123] that affect telecommunications signals, are not generally relevant or noticeable in packet delay calculations.)

To gain time so that whatever actions are necessary can be taken transparently to the user, the playout buffer, before playback, adds a fixed artificial delay ("jitter control time", or simply "control time", essentially some extra waiting time) to the first packet of every talkspurt. (A talkspurt is defined as a sequence of speech not broken by a pause or silence.) The size of this delay varies according to implementation and can be changed dynamically, within certain lower and upper limits, in order to adapt to network conditions [303].

Accurate sizing of a playout buffer is a challenging process, with two contradicting objectives: to replay all packets received from the network (which is not always possible, since extremely premature or excessively delayed ones are effectively lost), and to avoid imposing a prohibitively high control time to packets received within the set time limits. More specifically, if D is the maximum tolerable one-way delay for a voice sample and d_i the delay experienced by a voice packet travelling over the network, then the packet can be buffered for at most

$$t_B = D - d_i \geq 0$$

Whenever $t_B < 0$, the packet must be discarded as too late.

Playout buffer size calculations can be based on network measurements [274], stochastic analysis [19], or a combination of both [278], and their result is a fixed control time value, accurate on a percentile basis (e.g., 95% of arriving packets fall within the maximum delay constraints [83]). In a more empirical methodology, packet voice delay constraints, typical packet durations and the fact that usually a playout buffer must hold at least 2 packets to be of any use in error recovery [154], combined, imply average sizes of a few (5-10) packets. Playout buffers can usually shrink or expand dynamically, according to network conditions [286].

Synchronisation of packets inside the playout buffer is generally performed via either of the following methods [274], [303]:

- *Null Timing Information* does not use any timing data but instead adds a fixed delay, D, to the first packet, equal to the total network transit time (including queuing), D_t, plus the packet generation time at the sender, D_g, i.e. $D = D_t + D_g$. This approach is obviously vulnerable to losing the first packet; in addition, because of unpredictable delay variation, silence intervals between talkspurts are not reconstructed accurately.
- *Complete Timing Information* uses full timing information to calculate a uniform overall delay, D, as $D = D_t + D_r$, where D_r is defined by the receiver, according to the playout buffer algorithm used. Timing information is recorded in special fields ("timestamps"), which can be global (requiring sender-receiver synchronisation to a common clock), relative (recording relative time between consecutive packets), or delay-based (using "delay stamps" to indicate the delay accumulated by a packet during its transit through the network).

The recovery of timing information can be complicated due to the combined effect of silence, packet reordering, packet loss and delay variability in the network. Fixed initial delays and timestamps may be increased in all these cases, confusing calculations at the receiver, hence a timestamp per se is not adequate to reconstruct timing.

2.6 Network Impairments

A number of technical problems can be caused by the transmission network itself, the most frequent of which (mainly in wired communications) are discussed in the following sections.

2.6.1 Main Types of Impairments

Common factors affecting the performance of a packet voice network are congestion, echo, delay, and errors (mainly in the form of packet loss). These are interrelated, often in a sequential fashion: congestion increases delay, which makes echo perceivable, while packet loss levels are also higher in a congested network. In addition to these, other impairment causes can also appear occasionally, as discussed in the following paragraphs.

2.6.1.1 Congestion

The extra bandwidth consumed by packet voice applications can cause overloading (congestion) to the network (more accurately, to one or more of its switching devices), not only because of the need to accommodate voice traffic in addition to data, but also due to service requirements (e.g., the use of a better quality CODEC), or even packet error conditions (e.g., when a repair technique, such as FEC must be employed [288]). High levels of congestion can, in turn, lead to increased delay and, eventually, packet loss levels, causing overall service disruption, experienced as call blocking or denial of service by the end user.

Congestion detection and notification can be done either implicitly [210] or explicitly [130] by the network, depending on the communication technology used. For interactive traffic, like voice, conventional recovery methods such as flow control are not adequate, because of the excessive delays associated with them, so the problem is usually handled by end-user applications (e.g., the sender or the receiver may chose to drop packets or frames of audio to reduce bandwidth consumption [398]).

2.6.1.2 Delay

Delay is one of the most difficult to handle impairments of packet voice systems, causing confusion, double-talking, impression of an idle line and mutual silence among conversation participants, at higher values. As delay increases, the communication progressively adopts a half-duplex character, until it is finally disrupted.

Several factors contribute to the delay that a voice packet experiences in its journey from the transmitter to the receiver (Table 2.5). Because of high-performance hardware, pure processing delays are generally very low (in the order of nanoseconds or microseconds) and can therefore be ignored as negligible. The switching delay is the sum of deserialisation, processing (e.g., routing table lookup, header modification) and serialisation delay; this, plus time spent for possible gatewaying (e.g., signal processing, CODEC translation) and (input/output) queuing operations, accumulate to the total delay incurred by each node in the network. (Other causes, which can increase delays by up to tens of milliseconds, such as operating system scheduling

[154], [233] and modem performance [129], are case-specific and do not change the overall picture presented in Table 2.5.)

Parameter	Location	Variability	Indicative Value
Encoding (A/D conversion, framing, compression)	Sender	Fixed	10 ms - 40 ms
Silence Suppression	Sender	Fixed	< 10 ms
Packetisation	Sender	Fixed	20 ms - 80 ms
Serialisation	Sender	Fixed	1 ms - 10 ms
Transmission (propagation)	Network	Fixed	10 μs/mile
Switching	Network	Fixed	< 100 ns
Gatewaying (translation/transcoding)	Network	Fixed	1 ms – 40 ms
Queuing (predominantly input)	Network	Variable	50 ms-200 ms
Deserialisation	Receiver	Fixed	1 ms-10 ms
Playout buffer	Receiver	Variable	20 ms-100 ms
Decoding (decompression, D/A conversion)	Receiver	Fixed	< 10 ns
Other	Sender, receiver, or network	Fixed	Usually negligible

Table 2.5: Delay parameters

The total delay experienced by data traversing packet networks is not only high at times, but it can also vary significantly, due to phenomena like queuing, the non-deterministic behaviour of switching equipment, as well as fluctuations in the routing paths, particularly for connectionless services like IP. This delay variation (jitter) is present in all communication networks, but mostly in packet switching ones, where the factors mentioned above are accentuated. Typical jitter values vary according to the underlying carrier network: the GSTN has 20-30 ms [26], which is mainly a function of clock synchronisation mismatches or physical transmission distance; LANs have around 10 ms and WANs can exceed 100 ms, depending on topology and network technology employed; these, in turn, have a direct impact on bandwidth and congestion characteristics [137], [274].

Delay can be measured either on a one-way or (most commonly) two-way (round-trip) basis (e.g., using the "ping" or "traceroute" utilities in IP networks). Often, one-way is not equal to the half of two-way, but, on average, this is a reasonable approximation. There are several studies for the maximum amount of tolerable one-way delay in a voice network and different values have been quoted, typically 150 ms-300 ms, depending on network type and conditions such as jitter, echo and packet loss (Table 2.6). However, beyond the empirical value of 200 ms (one-way), the phenomenon starts becoming noticeable, talkspurt gaps emerge, interactivity is restricted, the communication increasingly adopts a half-duplex character and the perceived quality deteriorates rapidly [26], [44], [137], [268].

Value	Type	Description	Source
< 10 ms	Round-Trip	Echo nonexistent, or imperceptible (sidetone)	[26]
10 ms	Round-Trip	Echo possible in near-end circuits (e.g., PBX)	[26]
35 ms	Round-Trip	Echo possible in long distance circuits (e.g., PBX)	[26]
45 ms	Round-Trip	Echo cancellation necessary	[26]
50 ms	Round-Trip	Echo cancellation necessary	[176]
< 100 ms	One-way	Delay unnoticeable	[176]
150 ms	One-way	Delay unnoticeable	[176]
200 ms - 250 ms	One-way	Delay unnoticeable or barely noticeable	[176], [274]
300 ms	One-way	Delay noticeable but probably acceptable	[176]
400 ms	One-way	Delay noticeable and unacceptable	[176]

Table 2.6: Characteristic limits for delay

Controlling jitter and the maximum amount of one-way delay, so that they remain within the desired limits, becomes the main latency-related objective when designing a packet voice network. This approach has also a direct impact on echo performance.

2.6.1.3 Echo

Echo is the phenomenon resulting in the originally transmitted voice signal being fully or partially reflected back to the sender (talker echo), from where it can be reflected again to the receiver (listener echo) and even continue bouncing back and forth (singing echo), until it is completely faded. Talker echo is usually the most noticeable. The echo phenomenon is generally caused by electric equipment imperfections (e.g., handset crosstalk, 2-to-4 wire conversions inside switches and channel impedance mismatches), or by acoustic feedback at the receiver (e.g., from speaker to microphone). These causes produce energy leakages across communication paths, resulting in attenuated copies of the original signal being reflected for a number of times.

The perceptibility of echo depends both on the amount of round-trip delay (in milliseconds) and on the strength of the reflected signal (characterised by the ITU-T Talker Echo Loudness Rating, TELR [178] measure). It has been found that maximum tolerable one-way delay values in a voice conversation are significantly reduced when echo is added. As a general guideline, 45 ms [26] or 50 ms [178] round-trip delay in a circuit means that echo will be present, but operators usually employ echo reduction mechanisms as of 35 ms [138], to allow for special line conditions (Table 2.6). Interestingly, for speech there is a desirable sort of echo, the sidetone [138], which serves as feedback to the talker, verifying that the handset employed is functioning; this effect, however, must occur at nearly zero delay, otherwise it becomes separately audible.

2.6.1.4 Errors

Depending on the type of packet service (connectionless or connection-oriented) and due to a variety of reasons, such as equipment malfunction, LAN frame collisions, transmission problems and, most commonly, congestion inside the network, packet errors (duplication, reordering, or loss) may occur, resulting in degradation of voice quality or even communication disruption. The impact of packet errors is often related to other factors, like compression; for example, lost segments of speech can cause state drift problems in many CODECs, as certain parts of a talkspurt are more vulnerable to loss than others [210]. By contrast, speech is much less sensitive to bit errors as high as 1% [308], which cannot always be said for other types of voiceband data.

2.6.1.5 Other

Less frequent -and, hence, of minor impact- problems, like hardware (e.g., microphone) imperfections, blocking (i.e., no or excessively delayed dial-tone), call dropping because of equipment malfunction, or transmission channel errors, can occur, depending on the network conditions prevailing during communication.

2.6.2 Recovery Methods

A variety of techniques exist for alleviating the effects of network impairments, varying according to the particular problem being tackled.

2.6.2.1 Congestion

Following the "end-to-end argument" [324], contemporary communication systems tend to rely on the edges of the network to reduce the amount of overload traffic. Congestion detection and notification mechanisms depend on the particular technology employed, from implicit (e.g., in IP [210]) to explicit (e.g., in Frame Relay [130]).

2.6.2.2 Delay

The strategy for reducing delay is composite, due to the variety of contributing factors (Table 2.5). The easiest approach is to tune sender and receiver parameters (e.g., processing power, CODEC and bandwidth) so that fixed delays are minimised. The difficulties arise when it comes to the network, which is the most unpredictable component of the equation.

Network-related delays can be improved by traffic prioritisation inside switches (e.g., using separate queues for packet voice). Strict preemptive priorities can cause a denial of service experience to data users, thus non-preemptive strategies are better. Fragmentation is also a technique, which can be used if many large data packets exist, so that voice packets are not bottlenecked behind them. One can also try to minimise the number of intermediate hops in the network (e.g., during call setup); this, like prioritisation, is easier for managed backbones. Finally, the playout buffer at the receiver is a standard mechanism for alleviating the effects of delay variation caused by the network.

2.6.2.3 Echo

There are generally two ways to counter echo, based on reducing either the delay, or the magnitude of the echo signal. Delays, as discussed, can be decreased by using high-speed circuits and fast network switches, with lower average queuing times, whereas the echo signal can be directly manipulated by employing special devices.

Two main types of such devices exist, echo suppressors [179] and echo cancellers [180]; the former work by inserting a large amount of loss in the return path of the signal, while the latter subtract a delayed copy of the original signal from the signal flowing in the return path [26]. Echo suppression is an older and rather inferior technique, as it allows speech to pass in only one direction at a time; echo cancellation, on the other hand, adaptively removes echo, while maintaining full-duplex communication [389]. In general, echo removal devices can be deployed at any point in a network (terminals, PBXs, gateways). Although they are indispensable for medium to long distance communications, they can also introduce various clipping side-effects, which at times become noticeable by the communicating parties [139].

2.6.2.4 Errors

In the event of a packet error, the receiver (i.e., the receiving application) has to compensate by applying appropriate techniques, according to the severity of the situation. The receiver may, of course, ignore the error, but most commonly it will have to try to repair (actually, conceal [57]) the damaged voice stream, based on extra information that can be extracted from already (and correctly) received voice packets. Error repair methods, according to the origin of this extra information, are classified into *sender-based* and *receiver-based* [288].

The simplest sender-based repair technique is retransmission, which is only suitable for high-latency, non-interactive voice applications, and then only under certain limitations (small number of retransmissions, not very large delay), although there have been successful demonstrations of interactive packet voice under the same kind of restrictions [91]. Retransmission is also good for non-speech voice traffic (e.g., fax), within the boundaries of associated timeouts, if any. A more advanced alternative for non-interactive applications is interleaving [287], [288], where consecutive audio units (e.g., CODEC frames) are placed in different packets and properly re-sequenced at the receiver; in the event of packet loss, instead of a large gap at the received voice stream, a number of smaller gaps occur, which makes repair easier, although the overall latency is increased as a result.

The most effective sender-based techniques, however, suitable for both interactive and non-interactive applications, are based on some kind of Forward Error Correction (FEC) [214], [288], where extra information is transmitted along with the original voice packets, thus allowing the receiver to calculate the lost voice segments with varying amounts of accuracy. This extra information can be derived from binary calculations (e.g., XOR, or parity coding), in which case it is media-independent. Alternatively, audio units can be transmitted in multiple packets, so that one of the copies can be used for recovery, in the event of loss; this method, called redundancy, is media-specific and can use lower-rate encoding for the copies (Lower Bit-rate Redundancy, LBR), to conserve bandwidth at the potential expense of quality and complexity, as LBR techniques usually perform inferiorly to FEC [214].

A modification of this approach, called "embedded" or "layered" coding [26], [114], [137], [348] transmits two or more different versions ("layers") of the media stream in the same packet and/or over different channels (e.g., IP multicast groups); this allows the sender to exercise flow control, in order to assist the receiver gracefully adapt to a wide variety of network conditions (such as bandwidth fluctuations, congestion, delay and loss) by selecting the most suitable layer, usually at the expense of bandwidth overhead; this way, the method is successful in absorbing longer-term network variations.

	Sender-based			**Receiver-based**		
	Retransmission	**Interleaving**	**FEC**	**Insertion**	**Interpolation**	**Regeneration**
Complexity	Low	Medium	High	Low	Low to Medium	High
Bandwidth	High	Low	Medium to High	N/A	N/A	N/A
Delay	High	High	Low to Medium	Low	Low	Low to Medium
Performance	Low	Medium	High	Low	Low to Medium	Medium to High
Applications	Non-interactive	Non-interactive	All	All	All	All

Table 2.7: A qualitative comparison of various error handling techniques

Receiver-based error repair methods, the earliest to appear [389], attempt to conceal the effects of a packet error by creating substitutes for the lost packets just prior to replaying the voice stream, either separately, or in tandem with (complementing) sender-based error recovery information [288]. This insertion process may simply be, with increasing effectiveness, a replay of nothing ("splicing"), of silence ("silence substitution"), of white noise ("noise substitution"), or of the last packet received ("repetition"), in the place of the lost packet. More sophisticated techniques employ pattern matching in the received signal waveform, in order to reconstruct, via interpolation, the lost packet, yielding notably improved performance. Even better results can be achieved in certain circumstances (e.g., depending on the CODEC used) by mathematically regenerating ("predicting") the contents of the lost packet from a small number of packets that have arrived immediately before and shortly after the lost one, with or without fading to match energy levels.

A qualitative comparison of the most common packet error handling techniques is summarised in Table 2.7. It is important to note here that the effectiveness of all packet loss recovery techniques diminishes with increasing packet loss and packet loss burstiness; a typical loss percentage of around 15% has been quoted as a limit [210], [288], although higher figures have been reported, depending on time scales and under favourable network conditions [154].

In summary, the effectiveness of loss recovery methods depends on the particular loss pattern and is better in conditions of intermittent loss, rather than burst loss. Furthermore, receiver-based techniques are of limited performance and most suitable in low packet duration and low error rate communication. Therefore, combining a sender with a receiver-based mechanism is bound to be most effective in the majority of cases.

2.6.2.5 Other

Types of network impairments other than the above are usually rarer and equipment-dependent, so they are handled on an ad hoc basis.

2.7 Quality of Service (QoS)

A communications network is characterised by the Quality of Service, that is, the type of performance guarantees and service differentiation it can provide to its users. QoS can be viewed from two perspectives: the User (application layer) and the Network (network layer). Numerous methodologies for parameterising, provisioning and measuring both have been developed for packet voice networks (Table 2.8).

2.7.1 User vs. Network QoS

User QoS represents the voice quality offered to the end user through the packet voice application. Whereas for non-speech signals this is more of a binary issue (i.e., either the quality is good enough for the service to work, or not), for speech it is gradual and depends on each user's perception, which, in turn, is affected by both subjective (e.g., the environmental noise, business/home usage) and objective (e.g., encoding method) factors [113], [229], [285], [394].

Network QoS relates to the service level offered to voice traffic by the packet network itself. Although there are several different levels of QoS a network may offer, depending on user requirements and the technology employed, most implementations require that at least 3 service classes be supported for full-scale QoS provisioning [138], [257], [388]: *best-effort* (the network performance degrades with increasing load), *uncongested* (the network behaves as if it were under low traffic) and *guaranteed* (the network offers specific QoS parameters at user request). Clearly, Network QoS can diversely affect User QoS and is the most complicated part of the QoS equation.

User	**Network**
Identifiability	Availability
Intelligibility	Capacity
Interactivity	Delay
Fidelity	Jitter
Naturalness	Loss

Table 2.8: QoS parameters for packet voice

2.7.2 Provisioning QoS

QoS support is usually fairly easy to implement at a user level (e.g., by using sophisticated applications, high-rate CODECs and advanced hardware), so the effort is focused on implementing it inside the network, where impairments of various types (e.g., delay, jitter and packet loss) can significantly affect overall performance. In general, QoS is an end-to-end requirement directly affected by network impairments, so a "weak link" anywhere inside the network is sufficient to degrade it considerably.

A key QoS differentiating factor in packet switched networks is the distinction between connectionless and connection-oriented operation. Connectionless packet services are easier to install and deploy, have low call setup times, and make the most of statistical multiplexing, but generally suffer from lack of reliability and, to some extent, predictability. A connectionless packet network can lose, duplicate, or reorder packets, route consecutive ones over different paths and exhibit unstable behaviour in terms of delays, bandwidth availability and other QoS-related parameters. Unless sophisticated extensions are incorporated, the network relies on end-user higher layer services to cope with or alleviate these problems. Connection-oriented packet services, on the other hand, are reliable and capable of supporting service guarantees at various levels, at the expense of higher setup times, complexity and usage costs. In these networks, static (pre-configured) or dynamic (on demand) connections serve end-to-end user communication. The former are generally preferable in terms of complexity and performance, but may be difficult to scale (i.e., achieve full connectivity as the network grows beyond a certain point).

In general, fixed or low variability routes, especially for the traffic path, tend to secure better service quality [388]. Connection-oriented protocols excel at this but even connectionless paths can be semi-fixed (e.g., in the case where specific devices, such as gateways, must be part of the path all the time), via source routing. Hence, the connection-oriented approach is more QoS-friendly than connectionless ones, thus all proposed mechanisms for offering QoS over a connectionless network like IP, try to mimic or impose connection-oriented operation.

Irrespective of the method of operation, there are three main strategies for provisioning QoS in packet networks:

- *Policy.* QoS is improved in a "brute force" strategy, by exercising stricter call admission control (blocking new or clearing existing calls) and flow/congestion control, or by allocating more resources (typically capacity) to the network. This approach is the simplest to implement but, apart from potentially expensive, it is necessary anyway and cannot remove the need for special QoS mechanisms, because resource limitations will continue to exist for certain technologies (e.g., cellular networks) and traffic demand constantly increases. It is also inadequate, for the additional reason that it may lead to inefficient use of the network during idle periods, thereby defeating its statistical multiplexing nature.
- *Tuning.* Numerous methods exist for improving the performance of a network without significant upgrades. Aggregation of multiple bandwidth streams into a single one that occupies the sum of the individual bandwidths, better accommodates bursty traffic, like voice; special techniques, such as header compression [230], are available for slow links; proper network sizing is useful for mid-to-high loads; and traffic engineering techniques can be used to improve QoS in a variety of packet network technologies.
- *Signalling.* QoS support is native in certain communication technologies, like ATM, or can be offered as an add-on in others, like IP. In both cases, applications can request specific levels of service using signalling mechanisms (RSVP/DiffServ/MPLS [388], Q.932 [202], Q.2931 [204]), which typically attempt to establish an end-to-end path with improved performance characteristics (e.g., lower delay or packet loss), in a connection-oriented fashion. This approach provides for the highest QoS, at increased implementation complexity and cost.

Numerous methods exist for assessing the effectiveness of the above three strategies.

2.7.3 Measuring QoS

To assess the performance levels achieved by a network, certain QoS parameters can be measured, either individually, or collectively, and provide an overall score.

For User QoS, in addition to proprietary mechanisms developed by commercial voice equipment vendors for this purpose, ITU-T has standardised a number of subjective [194] and objective [196] methods for assessing the quality of voice signals. For speech, the most common such method is based on Mean Opinion Score (MOS) [195] measurements. MOS methods can produce varying (and, at times, controversial) results, depending on the test environment (software, hardware, topology, delay), participant characteristics (gender, education, or even temporal mood) and tested equipment (a classic example is the mobile phone service, where, because of lower expectations, MOS values tend to be exaggerated). In fact, MOS measurements produce the so called "listening MOS", which does not take into account delay, thereby creating the need for the calculation of the more accurate "conversational MOS" [214]. ITU-T specification G.107 describes a method (called the E-Model) for estimating MOS in the case of hybrid communication (i.e., for networks that use both circuit switching and packet switching), by comparing two reference connections involved [177]. Although the E-Model produces rather subjective results, it nevertheless is extensively used (as originally specified, or in variations) and forms the basis of many automated tools and techniques for assessing user QoS [156].

Additional tests may include other P.800 metrics like Absolute Category Rating (ACR) [194], Comparison Category Rating (CCR), and Degradation Category Rating (DCR). Another P.800 metric, the Perceptual Speech Quality Measurement (PSQM), initially developed by the ITU-T to test CODEC quality, calculates MOS scores after performing automated low-level measurements on the transmitted voice stream, and can successfully be used instead of employing human listeners. PSQM, however, is a metric developed and optimised mainly for stand-alone circuit-switched CODEC operation, and hence does not account for situations like packet loss, speech clipping and jitter [196]. An improved version, PSQM+, addresses some of these problems, but newer methods have also been proposed. One such method is the British Telecom (BT) Perceptual Analysis Measurement System (PAMS) [308]. PAMS provides automated voice quality scoring that yields results close to the corresponding subjective MOS measurements. Additional end-user QoS metrics, such as Grade of Service (GOS), Dial-Tone Delay (DTD) and Post-Dialling Delay (PDD), have been defined by the ITU-T [175], the latest being the Perceptual Evaluation of Speech Quality (PESQ) standard [197], which, unlike its predecessors, is able to predict voice quality in a very wide range of conditions, including CODEC distortion, errors, noise, delay and jitter. The Perceptual Evaluation of Audio Quality (PEAQ) [171] from the ITU-R can also be used for objective quality assessment.

For network QoS, measurements are typically performed on general parameters like delay, delay variation and packet loss, but there are more specific ones, such as the ITU-T Answer Seizure Ratio (ASR), Post Gateway Answer Delay (PGAD) and Average Length Of Conversation (ALOC) [174], created specifically for assessing gateway performance. Depending on the network technology involved (e.g., IP, FR or ATM), sophisticated tools exist that can be used for automatically calculating these or other similar parameters [156].

2.8 Network Technologies for Packet Voice

Numerous packet switching technologies have emerged since the early 1970s and some of them are still used for implementing contemporary networks. These include the connectionless IP, IPX and SMDS/CBDS, as well as the connection-oriented X.25, SNA, FR and ATM [363]. For packet voice transportation, in particular, at the desktop or in the backbone, the most popular solution is by far the Internet Protocol (IP) [297], with Frame Relay (FR) [130] and Asynchronous Transfer Mode (ATM) [258] still used in certain markets (e.g., for access and in the backbone, respectively). IP, FR and ATM have also been suggested as suitable for replacing the ubiquitous General Switched Telephone Network (GSTN) [394].

As discussed in Chapter 3, there are two major implementation approaches to packet voice over IP (VoIP) [128]: One uses the ITU-T H.323 [188] protocol suite for signalling, and the other uses the IETF Session Initiation Protocol (SIP) [318] protocol suite for the same purpose; both, however, rely on the Real-time Transport Protocol (RTP) [337] for media encapsulation. H.323 is older, rather complex and widely implemented, and by design reflects the needs of the telephony world, whereas SIP is simpler and more packet-oriented by conception, as discussed in Chapter 3. Since IP is a "best effort" connectionless protocol, a great amount of work in VoIP is dedicated to improving its native QoS disadvantage. On the other hand, the universal availability of IP networks (mainly due to the Internet) retains IP at the top of the list for packet voice services, with constantly increasing influence [63].

Frame Relay is a connection-oriented service, which has been successfully conceived as a replacement to X.25 [130]. The standard that has emerged in the late 1990s out of a variety of proprietary efforts to carry voice over Frame Relay (VoFR), is the Frame Relay Forum FRF.11

specification [116], which allows telephony-like services, with a wide variety of compression, silence suppression and channel multiplexing combinations. FR is superior to IP in terms of QoS, due its connection-oriented nature, its ability to guarantee a Committed Information Rate (CIR) of transmission and the explicit congestion notification mechanisms it incorporates; it also provides (in combination with a WAN technology, such as ATM) an easy migration path from private leased line voice networks, thus being attractive for many usage scenarios. However, its rather limited availability, positioning as an access technology and higher costs has not allowed it to become the first choice for packet voice.

ATM was chosen by the CCITT (now ITU-T) in the late 1980s as the technology that would give Broadband ISDN (B-ISDN) networks the ability to accommodate virtually all types of traffic on demand, at high speeds and with guaranteed QoS (in fact, ATM is the only packet switching technology that can offer QoS at the levels experienced in the circuit switched GSTN). This "multiservice" nature is implemented via the ATM Adaptation Layer (AAL), which has been diversified into 4 versions (AAL1, AAL2, AAL3/4 and AAL5) to cater for transportation of different types of media. Voice services (VoATM) have been defined for AAL1 (Constant Bit Rate/Circuit Emulation), AAL2 (Variable Bit Rate, for silence suppression) and AAL5 (Variable Bit Rate, carrying voice either directly, or encapsulated in other packet technologies such as IP and FR). Innovations like the small, fixed-size packet ("cell", 53 bytes long), permit very high switching speeds and its QoS guarantees make the protocol seem ideal for voice traffic, far more than IP or FR, but technical complexities, an inefficient standardisation process and the inevitable high costs of deployment as a result, have restricted ATM to the network backbone.

A number of other packet technologies, including Ethernet, Cable, Digital Subscriber Line (DSL) and wireless (e.g., Wi-Fi), have been suggested and used for telephony-level services [394]. However, these are mainly suitable for the desktop (client side), or destined to fulfil smaller parts of the packet voice puzzle, rather than becoming technologies of choice for large-scale implementations.

Key technical differentiating factors among IP, FR and ATM, as well as the other technologies usable for voice transportation, from a user perspective, are features (functionality), availability, reliability, security, interoperability, scalability, administration and cost (billing); from a network point of view, on the other hand, there are significant differences in terms of overhead, QoS, and access speeds (Table 2.9).

Property / Technology	**GSTN (TDM)**	**IP**	**FR**	**ATM**
High speed	✓	✓	✓	✓
Connection-oriented	✓	X	✓	✓
Multiplexing	✓	✓	✓	✓
Dynamic bandwidth allocation	X	✓	✓	✓
VBR	X	✓	✓	✓
Reliability	✓	X	✓	✓
Guaranteed QoS	✓	X	X	✓
Multiservice (by design)	X	X		✓
Minimum cost	✓	✓	X	X
Universal availability	✓	✓	X	X
Approximate OSI layer(s)	1 to 7	3 to 7 (TCP/UDP)	2	2, 3 and 4 (AAL)
Primary service type	Voice	Data	Data	Any

Table 2.9: IP, FR and ATM characteristics from the user perspective

Interestingly, all packet voice technologies can be or are indeed used as carriers for the Internet Protocol, so, apart from the usual link layer scenarios (e.g., IP over Ethernet, or DSL), there are several cases where VoIP is carried over FR, or over ATM (usually AAL5), with additional overhead and performance implications, since voice overlaid this way cannot make use of the native advantages of the underlying network (e.g., ATM QoS). This (expensive and complicated, but not rare) overlaying is perhaps the best proof for IP's dominance as a packet switching technology, in both the data and the voice worlds. Indeed, universal availability (by dialup, broadband, or faster connections), ease of implementation, and low cost, indicate the clear advantage of IP at the desktop (LAN) and justify the progressive displacement in its favour of FR at the access network (MAN) and of ATM at the backbone (WAN).

2.9 Conclusion

The advantages of packet switching over circuit switching have been progressively established as of the early 1970s, and already seen as commonplace since the previous decade. Due to these benefits, virtually all packet switching technologies have been considered for voice transportation, but from the mid-1990s IP has prevailed over theoretically superior technologies such as Frame Relay and ATM, due to its ease of implementation, low-cost integration at the desktop, and universal presence.

This chapter has provided a critical review of the main techniques used for overcoming the challenges associated with packet-based voice communication. As seen already, many of these techniques are common across all packet technologies (including IP, FR and ATM), and many of the solutions developed are still applicable (if not applied) to contemporary VoIP networks; thus, similar solutions are also used for the implementations presented in Chapters 5 and 6. The (several) additional technical issues related specifically to IP Telephony are discussed in the next chapter.

CHAPTER 3

IP TELEPHONY

3.1 Overview

Due to its simplicity, efficiency, flexibility, universality and low cost, the Internet Protocol (IP) [297] has established itself as the platform of choice for offering integrated multimedia communication over packet networks [245], [286]. A wide variety of Voice over IP (VoIP) technologies have been developed for implementing traditional telephony services, at various levels of success, and there is an ongoing effort for standardising such services in a global scale [72], [128], [253], [377].

This chapter provides a critique of the present state of the art in VoIP, addressing not only the most common areas such as signalling and media protocols, but also less investigated ones, like security, interoperability and pricing; emphasis is also given on discussing the problem of scalability, which is core to this thesis, and particularly on the currently prevailing softswitch architecture, upon which the framework presented in Chapter 4 (and, more generally, this work) expands. The degree of detail varies according to the extent each of the technologies described is used or referred to later on in the thesis. Importantly, many of the standards and the techniques overviewed here have been designed for, and are directly applicable to, other types of media such as video, therefore much of the following discussion is relative to the more general subject of IP Telephony (i.e., VVoIP and additional multimedia services).

3.2 Architecture of a VoIP Network

A Voice over IP network operates as an application-layer overlay of an IP-based infrastructure and is implemented by three main components, the *Terminal*, the *Signalling Server* and the *Gateway* (Figure 3.1). These components can appear in different varieties, according to the particular subset of functions they implement each time, but their basic role in the network remains as described next.

3.2.1 Terminal

The terminal acts as the signalling and media interface to the network, allowing users to place and receive calls, and exploit other voice-related services, but additional types of media, such as video and instant messaging, may also be supported. Signalling operations include call control and capability negotiation with the other side, while typical media functions are encoding/decoding, packetisation, synchronisation, silence suppression, as well as error detection and correction.

A terminal used for speech calls can be realised as a software program ("softphone") running in a computer or similar device with audio capabilities, as an intelligent stand-alone phone with IP connectivity ("IP Phone"), or often as a combination of legacy equipment (e.g., an analogue phone) with an adapter that interfaces it with the VoIP network [159]. More sophisticated devices, such as gateways, media mixers and conferencing units, also act as terminals in the sense that they receive and place calls as communication intermediaries. For other types of voiceband data (e.g., DTMF, fax or modem signals) the role of the terminal is assumed by the corresponding devices, either in hardware, or in software.

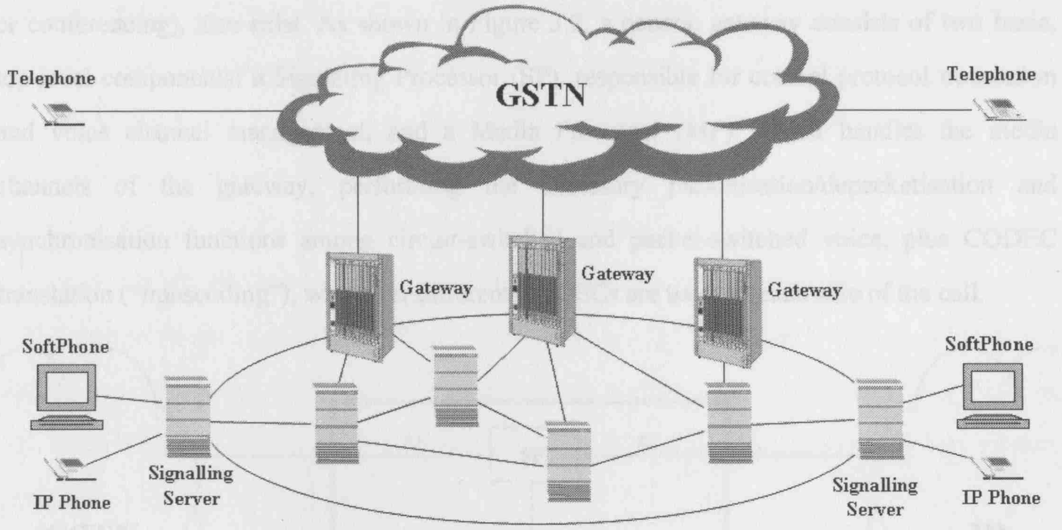


Figure 3.1: Architecture of a VoIP network

3.2.2 Signalling Server

The signalling server is the device that performs call management inside the network, including setup, routing and clearing of calls, admission control, address translation, location registration for mobility, maintenance of Call Data Records (CDRs), Authentication, Authorisation and Accounting (AAA), and billing, wherever applicable. A special type of signalling server is the *application server*, i.e., a device that implements certain calling features in the network (e.g., call parking or forwarding), similar to the SS7 [198] and IN [120] models.

Depending on the particular signalling protocols involved, these functions can be integrated in the same physical device, or distributed across a number of different devices, all communicating over IP. The presence of a signalling server is optional for smaller scale communication (e.g., over a private LAN or in non-commercial point-to-point calls), but mandatory for large-scale networks.

3.2.3 Gateway

A gateway is a logical device that offers connectivity to an IP network and another type of network, often an SCN like the GSTN [316]. It is typically used for interfacing VoIP "islands" with other, incompatible voice networks (e.g., the GSTN or a VoIP network using different protocols) via both signalling and media translation [394], and it does so by operating as an endpoint for both sides of each connection, as seen in Chapter 2 [315]. Special types of gateways, implementing part of the functionality (e.g., only signalling or only media translation, or conferencing), also exist. As shown in Figure 3.2, a generic gateway consists of two basic, top-level components: a Signalling Processor (SP), responsible for control protocol translation and voice channel management, and a Media Processor (MP), which handles the media channels of the gateway, performing the necessary packetisation/depacketisation and synchronisation functions among circuit-switched and packet-switched voice, plus CODEC translation ("transcoding"), whenever different CODECs are used on each side of the call.

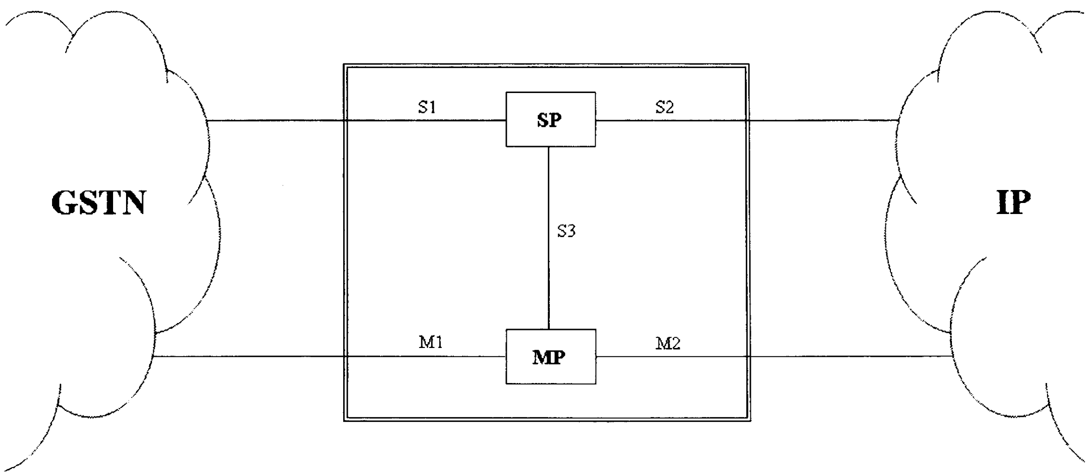


Figure 3.2: Top-level diagram of a generic VoIP gateway

Two types of interfaces can be found in a VoIP gateway: *External interfaces* (such as the S1, M1 and S2, M2 pictured in Figure 3.2) connect the gateway to the VoIP network and, depending on the operational scenario, can be either circuit or packet-based. The circuit-based ones will usually terminate GSTN signalling (e.g., Q.931 [201] or SS7 [198]) or media (e.g., TDM voice), whereas their packet-based counterparts will be used to carry IP signalling or media packets. *Internal interfaces*, on the other hand (such as the S3 shown in Figure 3.2), provide for connectivity among the gateway components and are essentially signalling channels, used for the communication of the various components of the gateway.

Contemporary VoIP standards [188], [318] have adopted the ETSI TIPHON model [107], which conceptualises a VoIP gateway as a distributed ("decomposed") device, where the SP is implemented as a combination of a telephony *Signalling Gateway* (SG) and a packet *Media Gateway Controller* (MGC, essentially a signalling server that is also referred to as the *Call Agent*), whereas the duties of the MP are undertaken by a *Media Gateway* (MG) component. The partition of the SP into SG and MGC components is not necessary from a technical point of view, but it facilitates the enforcement of administrative policies, as it allows large GSTN providers to hide the internals of their SS7-based networks via operating the SG and providing external access to third-party gateway operators through it, over IP [79], [146].

More specifically, depending on desired functionality, each decomposed gateway consists of a single MGC, zero or more SGs and zero or more MGs. The SG extracts control information and relays it to the MGC, while the MG focuses on media interworking and can vary in nature from a single-user IP phone to a multi-rack device, collectively supporting tens of thousands of GSTN and IP media ports. The MGC manages the media paths flowing through a decomposed gateway by exerting control over its MGs in a master-slave fashion, using stimulus signalling. All "internal" (SG, MGC, MG) gateway communication takes place over IP-based control protocols.

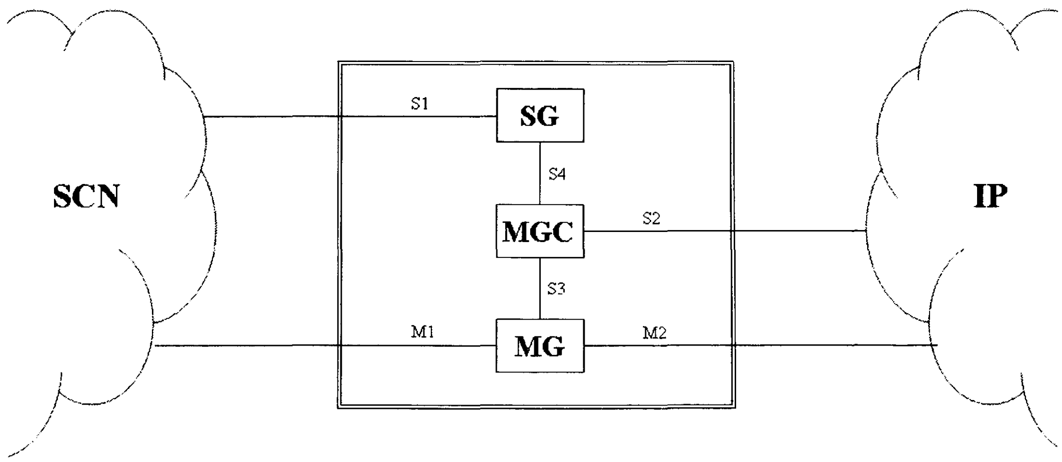


Figure 3.3: Top-level diagram of a decomposed VoIP gateway

The decomposed gateway has the same type of external interfaces as the generic gateway shown in Figure 3.2, but internally it differs in two aspects: first, there is an extra interface, S4, used for the MGC-MG communication (Figure 3.3). Secondly, all internal interfaces are IP-based, which allows the various components to be spaced afar, provided that the performance targets required by the protocols used (mainly delay behaviour and reliability) can be met. The exact IP-encapsulated application-layer protocol carried over internal links depends on the functionality implemented by the MGC each time, however the most typical cases are the MGC-SG communication (interface S3 in Figure 3.3), for which the IETF is developing the SIGTRAN [282] family of protocols, and the MGC-MG communication (interface S4 in Figure 3.3), which is standardised around the MGCP [5] and H.248/MEGACO [136], [187] protocols. The H.248/MEGACO protocol can also be extended for controlling monolithic, non-decomposable devices [80].

In general, decomposed gateway architectures are very important in building scalable IP Telephony networks, for a number of reasons [61], [66], [79], [80], [85], [89]. For a start, they can accommodate larger numbers of calls, due to their distributed operation. Moreover, they offer increased reliability by allowing redundant components (multiple MGCs or MGs) to be deployed. Third, because they use standardised protocols for internal and external communication, they facilitate the cooperation equipment source from multiple vendors, thus increasing competition and reducing implementation costs.

3.2.4 The VoIP Protocol Stack

IP Telephony functionality is implemented through a combination of protocols and accompanying specifications. The parts of the TCP/IP stack which are related to VoIP, in particular, can be roughly classified into 3 categories (Figure 3.4): *signalling protocols* (including H.323 [188], SIP [318], H.248/MEGACO [136], [187] and TRIP [314]), *media protocols* (RTP [337] and the various CODEC standards [229]) and *cooperating protocols* (all the remaining general-purpose ones, which are not directly related to IP Telephony - e.g., the classic Internet protocols such as TCP [298], UDP [296] and IP [297]). As shown in Figure 3.4, core VoIP protocols (i.e., those belonging to the first two categories) are (or can be considered to be) parts of the application layer in the 5-layer TCP/IP reference model, although a 4-layer IP Telephony reference model has also been proposed, where such protocols operate in more than one layer, depending on the communications scenario [253].

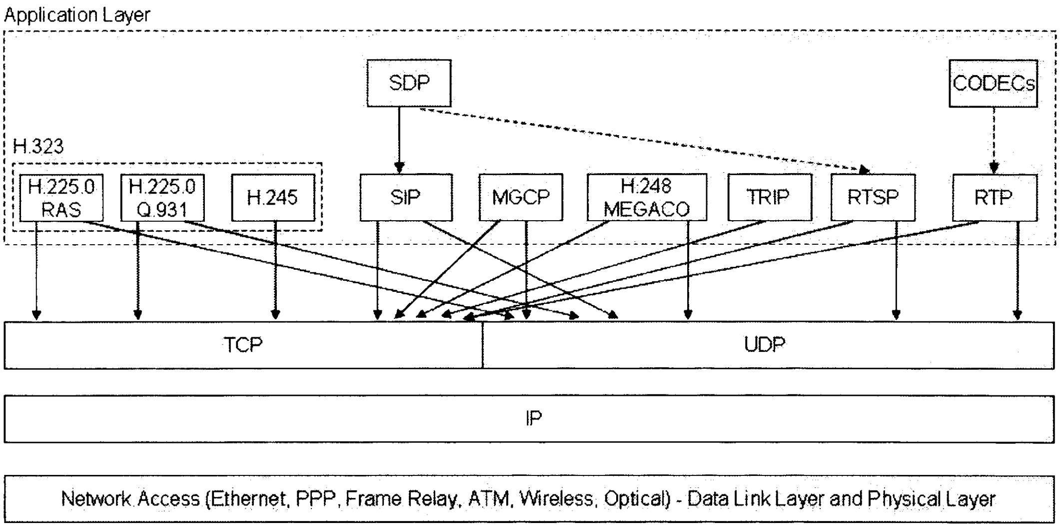


Figure 3.4: The VoIP protocol stack

3.3 The Control Plane

A number of protocols comprise the VoIP control plane, which is dominated by call management functions, including call setup and release, as well as the exchange of control information (e.g., status) during a call. Call setup, in particular, is implemented in three phases: *registration*, during which the calling terminal makes itself known to the signalling server and is granted permission to place a call; *notification*, during which the actual messages used to discover and notify the callee are relayed from the calling terminal to the called terminal through a path of signalling servers; and *connection*, during which the communicating terminals negotiate a common set of capabilities and allow the users to start exchanging actual media. For call release, the same procedure is followed in the reverse order, with the exception of the (un)registration, which is usually not necessary on a per-call basis.

Either H.323 [188] or SIP [318] is used for general call management in IP Telephony networks, and MGCP [5] or H.248/MEGAGO [136], [187] specifically for Media Gateways. All these protocols impose certain performance and reliability requirements, therefore they are either run over TCP, or enhance standard UDP behaviour with their own retransmission mechanisms to circumvent packet loss.

3.3.1 H.323

H.323 [188], [235] represents the ITU-T approach to IP multimedia signalling. It is an application-layer, binary communication protocol, which uses ASN.1 notation [207] for message encoding and interpretation. H.323 actually is an umbrella specification that coordinates numerous control and user plane communication protocols.

Four main entities can be identified in an H.323 network. H.323 *terminals* vary in capabilities such as media and encoding formats supported. The role of the signalling server is assumed by a control device, the *gatekeeper*. Furthermore, *gateways* offer interoperability with other types of multimedia networks (e.g., PSTN, ISDN, GSM, or even H.323 and SIP-based ones), and a special type of them, the *Multipoint Control Unit* (MCU), consisting of one *Multipoint Controller* (MC) for signalling and zero or more *Multipoint Processors* (MPs) for media traffic, handles centralised conferences; decentralised conferencing is also possible via IP multicasting, using an MC but no MPs. Terminals, gateways and MCUs can place or receive bidirectional, real-time calls and are thus collectively referred to as *endpoints*. Several H.323 endpoints controlled by the same gatekeeper constitute a *zone*, in which additional gatekeepers may exist only for secondary purposes (e.g., load balancing and redundancy). Endpoint addressing is implemented via *aliases*, which can have numerous formats, ranging from e-mail (e.g., auser@the.net) to E.164 identifiers (i.e., telephone numbers) [172]. Address translation is performed either locally, via manual configuration, or, in the general case, automatically by the gatekeeper itself.

Call management in H.323 networks is conducted in three main phases, in a master-slave fashion, with necessary control information (e.g., destination signalling IP address and port) offered successively, on an as-needed basis. First, H.225.0-RAS (Registration, Admission and Status) [184], run over UDP, allows endpoints to discover a gatekeeper via broadcasting or multicasting, register with it and periodically refresh their registration to avoid expiration, get permission to participate in calls, request a particular QoS level (like a certain amount of bandwidth), and receive user location (mobility) services. RAS messages are classified as Request, Response, Command and Indication, most frequent of which are the first two, usually in the format xRQ (Request), xCF (Confirmation) or xRJ (Rejection), where "x" is replaced by a capital letter indicating the actual message (e.g., ARQ denotes an Admission Request). H.225.0-RAS is followed by H.225.0-Q.931 [184], over TCP, implementing the main part of call setup, either through a gatekeeper (gatekeeper-routed signalling), or straight between communicating endpoints (direct signalling). In both cases, a variation of the ISDN Q.931 standard [201] is employed, using a subset of its messages, modified properly (particularly by customising the User-to-User Information Element) to accommodate packet-related information, such as the transport address (IP address and port) of the callee.

Once the call is connected via H.225.0-Q.931, H.245 [186] takes over to perform capability negotiation between the communicating endpoints over TCP, in order to setup two unidirectional *logical channels*, i.e. exchange a pair of destination transport addresses to which voice packets will be sent on either direction. To resolve any conflicts that may occur during the capability negotiation process, one of the two endpoints is designated as the master, according to a simple election process based on the characteristics of each endpoint (for example, MCUs have higher priority than plain terminals). Eventually, either of the communicating endpoints will terminate the call and a brief exchange of appropriate protocol messages, following the reverse sequence (i.e., H.245, H.225.0-Q.931 and H.225.0-RAS), will inform the network.

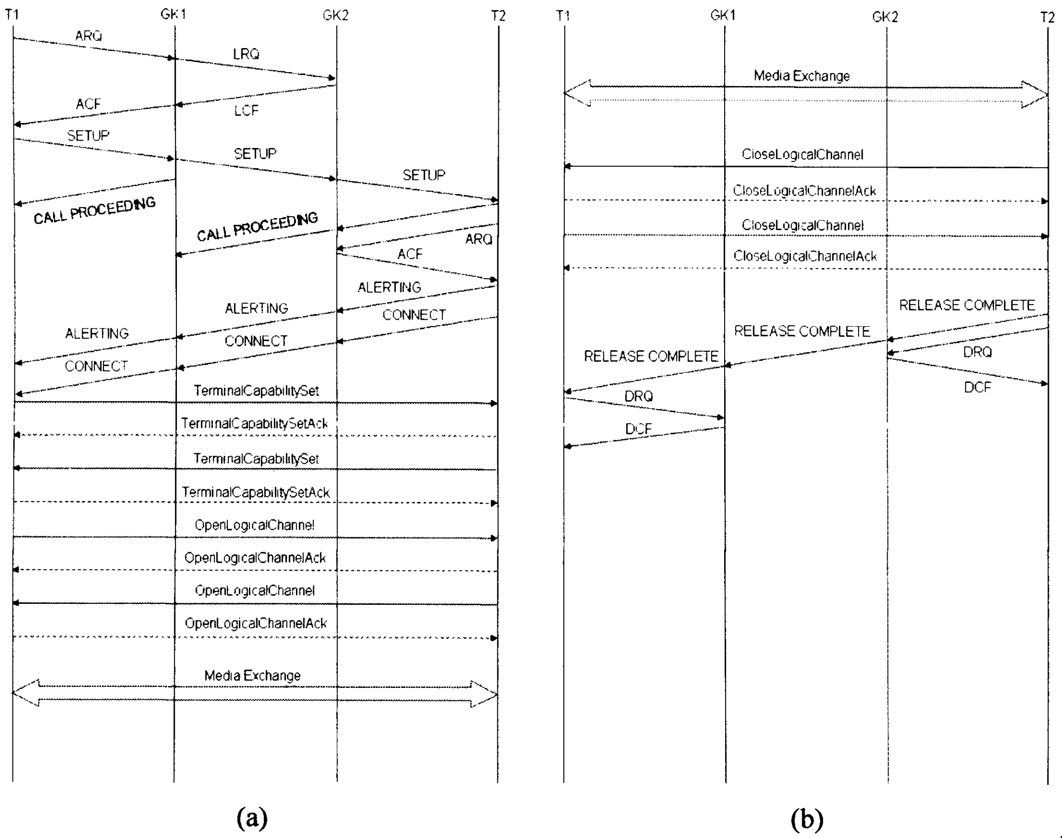


Figure 3.5: A simple H.323 call: (a) call setup (b) call release

A simplified example of this message sequence is shown in Figure 3.5, which demonstrates a point-to-point communication taking place across two different H.323 zones, between two terminals, T1 and T2, associated with the corresponding gatekeepers, GK1 and

GK2, that route call signalling [188]. The user of T1 places a call by supplying the T2 alias to GK1 and asking for admission. GK1 sends a location request (LRQ) to GK2, for address translation, and then grants T1's request, so that the latter can proceed with an H.225.0-Q.931 SETUP message. This is transmitted across to T2 (with the appropriate local acknowledgements in the form of CALL PROCEEDING), which must get permission from its local gatekeeper before starting to ring. When that happens, an ALERTING message is transmitted back to T1 and, soon after, the user at T2 accepts the call. This leads to an exchange of terminal capabilities and then of requests for unidirectional logical channel establishment, the successful completion of which means that the actual media exchange can take place, directly between the two terminals. Later on, the T2 user releases the call, which results in a sequence of events taking place in the inverse order: the logical channels are closed via H.245, an H.225.0-Q.931 RELEASE COMPLETE message is sent back to T1 and, finally, both terminals send a disengage request (DRQ) to their corresponding gatekeepers, so that the latter are notified accordingly. Depending on implementation, during gatekeeper-routed call signalling (as in this example) the H.245 messages may or may not be routed through the gatekeeper [128], [159].

The full H.245 message sequence, as originally appeared in 1996, in version 1 of H.323 [159], presents numerous operational challenges. For instance, it makes it difficult for pre-recorded, network-originated voice messages to be played in the event of an extraordinary condition (e.g., the user typing a wrong telephone number); also, the sheer number of packets exchanged, combined with TCP operation, build up excessive delays in establishing the media path, often resulting in no media flowing for several seconds after the call is connected [159]. For this reason, simplifications have been developed in subsequent versions of H.323 (particularly since version 2, which was a major upgrade), such as the possibility of running H.225.0-Q.931 over UDP (with retransmissions), as specified in Annex E of the standard, and the ability to open bidirectional logical channels, thus halving the number of H.245 messages exchanged. Two more significant improvements reduce latencies by allowing capability negotiation to be embedded in the H.225.0-Q.931 phase: "Early H.245" achieves so by allowing the inclusion of the H.245 transport address in the SETUP or CALL PROCEEDING messages, whereas "Fast Connect" allows the establishment of bidirectional logical channels immediately after the initial SETUP message has been sent, via the inclusion of a special "fastStart" Information Element.

H.323 was designed with GSTN interoperability in mind, and thus incorporates protocol translation and feature transparency both at the edges and in the backbone of the network, via special translation functions (typically VoIP gateways). The ITU-T H.323 documentation and several additional specifications detail the interworking of H.323 with Q.931 [201] (GSTN access), SS7 [198] (GSTN backbone signalling, particularly ISUP [200]) and QSIG [101] (PBX interconnection). Although the entire GSTN functionality is not yet found in H.323, the common feature set of these two network technologies is well-supported.

Overall, ten years after the publication of its first version, H.323 can be considered a stable and mature protocol. Initially of limited scope (LAN only) and high complexity, trying to cover all possible interconnection scenarios and offer compatibility for many different types of (now obsolete) telephony equipment, the standard has been progressively simplified and strengthened, taking advantage of the standardisation of packet-based multimedia around IP. Its current form, Version 5 [188], published in 2003, adds only minor improvements to the previous version of early 2000, thus further justifying the maturity hypothesis. As a result, H.323 remains the most widely implemented signalling standard for VoIP [159].

3.3.2 Session Initiation Protocol (SIP)

SIP [318] was developed within the IETF, for signalling in IP multimedia networks. It is an application-layer, text-based communication protocol, which uses ABNF [84], an augmented version of the BNF notation, for message encoding and interpretation. SIP follows the client-server, request-response model of HTTP [110], and is capable of establishing, managing and terminating calls that involve two or more parties and a multiplicity of media types (including - but not restricted to- voice). It can run over either TCP or UDP, but is independent of specific layer 4 protocols, having its own, retransmission-based reliability mechanism for unreliable transport connections. A number of companion IETF standards are used to complete the full call management set of functions needed in a SIP-based network.

SIP messages are classified as requests or responses, are encoded in plain, case-insensitive text (in ISO 10646-compliant format) and have two parts, the header and the body. The header follows the HTTP syntax rules and consists of several "lines", providing such information as source and destination of the call, route followed, encryption method used (if any), and length of the body part. The body part itself describes the media and the association ("session") existing between the communicating parties, and is usually encoded in Session Description Protocol (SDP) [150] format.

Numerous entities can be encountered in a SIP network. Terminals (voice or other) are called *User Agents* (UAs), consisting of a User Agent Client (UAC) component capable of placing calls, and a User Agent Server (UAS) component responsible for accepting calls. A *Back-to-Back User Agent* (B2BUA) receives requests as a UAS and posts requests as a UAC, maintaining full state. The main signalling server is the *Proxy Server*, which acts as a call router, on a hop-by-hop basis, similarly to SMTP [227]; in addition, it plays the role of both an (intermediate) client and server for calls, enforces domain policies and hosts other signalling functions, such as Authentication, Authorisation and Accounting (AAA). A *call stateful proxy* maintains full state about a call (i.e., keeps track of all messages exchanged), a *transaction stateful proxy* focuses on a particular transaction, whereas a *stateless proxy* simply acts as a relay of signalling messages, without any record kept.

A number of additional server entities are defined by SIP: a *Redirect Server* is a special case of a UAS that switches a call to alternative destinations (e.g., in the case of a mobile terminal) by returning routing information to the calling UA (or the calling proxy server), which can then re-initiate setup to the new destination indicated; a *Registrar Server* records the current location of a user; finally, a *Location Server* is a non-SIP entity that stores information about a callee's possible locations and can provide mappings to actual addresses on request from a redirect or registrar server; location server databases are updated by registrar servers (acting as front-ends to the database), or by another protocol, such as LDAP [384]. In actual implementations, redirect and/or registrar servers are often collocated with proxy servers. SIP networks also use *Gateways* for interoperability with non-SIP networks (e.g., the GSTN, or H.323 networks). Conferencing can be centralised or decentralised, but no special entity is specified for that purpose. Addressing is implemented via special Uniform Resource Identifiers (URIs), which are similar in format to e-mail addresses and typically constructed from a username and a hostname. Address translation is performed either locally, via manual configuration, or, in the general case, automatically by a SIP server, usually with the help of a location server.

Basic call signalling in SIP-enabled networks is described by the protocol itself [318], which provides for registration and status, call setup and capability negotiation, via its various request methods. A request and its associated responses is treated as a single *transaction*, each unique peer-to-peer communication between two UAs that lasts for some time is called a *dialog* and a media communication during a dialog constitutes a *session*; for instance, two UAs engaging in a videoconference dialog would typically create two sessions, one for audio and one for video, and possibly additional ones, such as a whiteboard-based collaboration. Session information necessary for capability negotiation (e.g., supported media types and CODECs) is carried in the body of appropriate SIP messages, which is usually formatted using SDP; however, SIP does not preclude the use of other session description mechanisms – suitable, for example, for initiating online gaming sessions.

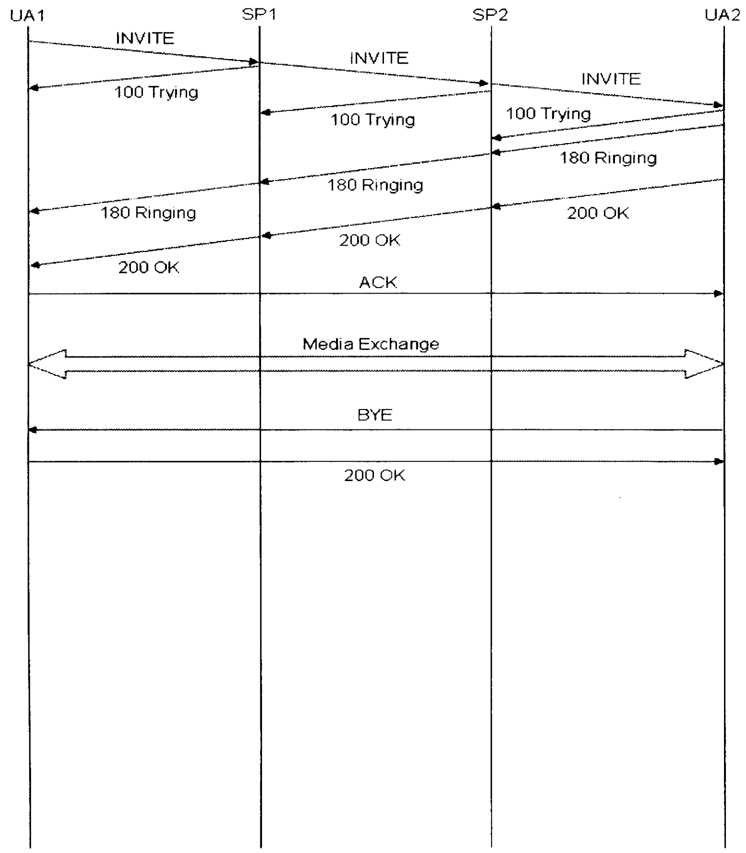


Figure 3.6: A simple SIP call; call setup and call release

A simplified example of call setup is shown in Figure 3.6, which demonstrates a point-to-point dialog taking place across two different SIP domains, between two user agents, UA1 and UA2, associated with the corresponding proxy servers, PS1 and PS2, that route call signalling [216]. The user of UA1 places a call by sending a setup message (an INVITE request) containing the callee's address URI and other necessary information, to SP1, the address of which it knows by manual configuration. SP1 replies with a "100 Trying" message indicating that it is proceeding with the call and forwards the INVITE to SP2, which produces a similar response to SP1 and routes the INVITE to the UA2, the actual IP address of which it has discovered by a means not shown in Figure 3.6 (e.g., by contacting a collocated location server). The T2 may produce a "100 Trying" response itself, just prior to instructing its UA to start ringing, a situation represented by the "180 Ringing" response, routed all the way back to the

UA1 through the same path (as recorded in the header of the original INVITE message). Eventually, the user at UA2 answers the call and a confirmation, in the form of a "200 OK" response, indicates to the UA1 that the media channel can be opened, since the capability negotiation has already been conducted as part of the INVITE transaction. A new transaction, indicated by the direct transmission of an ACK request from the caller to the callee, concludes the call setup process. Later on, the user at UA2 releases the call, resulting in a SIP BYE request to be sent directly to UA1, which replies with a "200 OK" message, thus completing the call release phase.

Following IETF's distributed design philosophy, signalling server functions beyond those related to basic call setup are not incorporated into SIP and are instead the task of other protocols. Admission control, for instance, can be implemented by extending QoS protocols such as RSVP [397], or policy-based mechanisms, like COPS [99], [147]. Call Data Records (CDRs), Authentication, Authorisation and Accounting (AAA), and billing will most probably be the responsibility of a call stateful proxy server, whereas network-level QoS will be applied by other, possibly lower layer devices (e.g., IP routers running appropriate protocols). More generally, SIP proposes custom mechanisms for the implementation of extra services, including the Call Processing Language (CPL) [242], CGI scripts [241] and Java servlets [U46].

Interoperability with the GSTN is a primary concern in the SIP development effort. As the protocol continues expanding to accommodate more of the features found in the GSTN, interworking is also investigated and a number of standards for interfacing SIP with Q.931 [201], SS7 [198] (including ISUP [200]) and QSIG [101] already exist [291], [292], [378]. This functionality is offered by special translation devices (typically, VoIP gateways) positioned at the interconnection points between GSTN and VoIP networks - however, beyond the basic call setup procedures, more complex interworking scenarios require further development work before they can be considered finalised [77], [159].

Between 1996, when the first, incomplete SIP draft appeared in the IETF, and 1999, when the protocol was standardised in RFC 2543 [53], a lot of progress was achieved. SIP was enriched in functionality from a simple call setup mechanism to a modular specification, offering additional facilities that range from instant messaging [87] and presence [88], to stimulus signalling and special provisions for telephone devices [378]. The protocol is constantly expanding in two directions: one, accommodating all traditional telephony (GSTN) features, for interoperability purposes; and two, supporting new applications of IP multimedia, such as the ones already outlined. This perpetual upgrading effort has the side-effect that it significantly increases implementation complexity and directly affects the protocol's stability as a common standard. That being said, SIP maintains its momentum as a favourite for next-generation IP multimedia signalling, due to its *versatility* (allowing it to constantly accommodate new applications, such as instant messaging [56], wireless VoIP [81], new IP-based networked multimedia architectures [54], [294], or third and fourth generation (3G and 4G) mobile [390]), *extensibility* (usually much easier for a text-based protocol) and, of course, *IP-friendliness* (by design).

3.3.3 H.323 vs. SIP

Numerous comparisons between H.323 and SIP, either as a whole or regarding certain features like service implementation, have been conducted over the past decade [125], [126], [219], [335]. Based on these studies and on the current status of the two protocols, certain comparisons can be drawn between them.

For start, it should be noted that, although both protocols first appeared in 1996, H.323 has enjoyed longer and more intense preliminary work. Therefore, a better landmark for comparison is year 1999, when SIP was first standardised as RFC 2543, while simultaneously H.323 reached Version 3, after which only minor updates have been made. This is an indication of superiority in terms of maturity on behalf of H.323 and, indeed, market penetration and interoperability across different vendor products seem to verify this claim [159].

More specifically, H.323 presents numerous technical advantages compared to SIP:

- **Capability Set**. H.323 is a fully featured "umbrella" specification, which replicates most functionality found in a telephony network, while remaining an IP-based standard. SIP, by contrast, depends mostly on extensions for offering the same set of features and is still incomplete in this respect.
- **Integration**. H.323 specifies call setup, call control and media handling. SIP focuses on call setup (session initiation) and depends on additional protocols for the other two procedures.
- **Interworking**. H.323 is considered superior to SIP when it comes to interoperability with the GSTN. This is partly due to the compatibility of protocols like H.225.0-Q.931 with corresponding GSTN standards such as Q.931 and SS7, and partly due to the design of the protocol, which reflects long telephony experience and has included interworking as a primary objective.
- **Efficiency**. H.323 is a binary protocol, using the Packet Encoding Rules (PER) of ASN.1

for the encoding of its messages and data structures. This approach requires higher effort for prototyping a system but, once this phase is completed, existing tools for automatic code generation can be used and message sizes can be kept small, resulting in lower bandwidth consumption. SIP, on the other hand, is text-based, meaning larger and more variable size messages, even if the "compact" (abbreviated) header mode is employed.

- **Maturity**. H.323 has been field proven and thoroughly revised during the past decade. The protocol is widely implemented in commercial (and some open source, like Open H.323 [U34], [U35]) products, both at end user and carrier level, backward compatibility and interoperability among different vendors is seamless for the core functionality and a large set of supplementary services is well-defined. SIP also enjoys universal support but lacks in implementation levels and depends on third-party proposals for most functionality beyond the lightweight core specification, an approach that leads to proprietary extensions and potential incompatibilities. As a result, SIP has to continue expanding, in order to achieve support for the entire feature set necessary for large-scale IP Telephony, encompassing the current GSTN, thus inevitably lacking in maturity compared to H.323.

On the other hand, SIP can be considered superior to H.323 in the following aspects:

- **Simplicity**. Because of its text-based message structure and the smaller number of messages required to establish a basic call between two endpoints, it is easier to rapidly prototype SIP products, compared to H.323. This advantage is further strengthened by the open architecture of the protocol, which has allowed the development of many public domain implementations, like the SER Proxy Server [U45]. Debugging of signalling exchanges can also be simpler, since messages are directly readable off the wire.
- **Flexibility**. SIP has been created with extensibility in mind, following the IETF philosophy of having a different protocol for each well-defined set of core functions. The text-based message structure, the tolerance to unknown message parts or types by both clients and servers, and the modular structure of the protocol, are products of a flexible design, which is easier to extend than a fully integrated standard like H.323. This openness is a particularly powerful and future-proof characteristic of SIP, however, as already discussed, it also slows down its maturation and can cause interoperability problems among different products or vendors, since extensions to the core specification appear frequently without necessarily conforming to the set guidelines [317].
- **Service Implementation**. SIP proposes well-defined mechanisms such as the XML-based CPL [242] for the implementation of services additional to basic call setup, whereas H.323

introduced a service creation architecture as of Version 4 and even that is still not complete. SIP is also superior in terms of integration with IP applications (e.g., web browsing, instant messaging, unified messaging), due to its IP-based design and mentality.

- **Call routing**. SIP follows the SMTP hop-by-hop routing model [227], based on proxy servers relaying signalling messages end-to-end, with explicit loop detection algorithms. This approach is probably better for larger scale networks than the H.323 model, which includes limited gatekeeper-to-gatekeeper communication facilities and implements a sub-optimal loop detection mechanism. Furthermore, SIP is proposed for communication among MGCs [128], thus it also has a strong potential for being increasingly involved in call routing to and from the GSTN [378].

There is an ongoing debate between the ITU and the IETF worlds as to whether H.323 or SIP will prevail in the race for the single IP multimedia signalling framework, and bias in favour of one or the other protocol remains a frequent phenomenon. However, an objective analysis of technical characteristics and market trends, as above, indicates that neither standard is ideal for all purposes, or can supersede the other. H.323 is superior for its maturity, interoperability and vendor support, whereas SIP excels in simplicity, rapid product development, flexibility and IP integration, thus being more fashionable among vendors for future IP Telephony network deployment [67], [332]. That being said, the two specifications seem destined to co-exist in the foreseeable future [63] and could only be obsoleted by a common successor "superprotocol" [133], offering a superset of their combined characteristics, after several years [53], [159], [235]. In the mean time, solutions (essentially gateways) that offer interoperability between the two protocols are already commonplace [162], [333], [345], [U1].

3.3.4 The Real Time Streaming Protocol (RTSP)

RTSP [334] is an application-layer signalling protocol for remotely controlling the delivery of multimedia content, on-demand, either in real-time (e.g., in an online "live" concert), or off-line (e.g., from a stored media repository). It was developed for use in streaming media applications, where one or more clients request one or more multimedia flows (e.g., a video and several audio streams, for a movie with stereo sound) from a remote server, over IP. While not directly used in interactive VoIP communications, it can be seen as a member of the same protocol family and complies with a similar operational framework.

Like SIP, RTSP messages are classified as requests or responses, are encoded in plain, case-insensitive text (in ISO 10646-compliant format), and have two parts, the header and the body. The header consists of numerous "lines" conveying information about the particular streaming session the message refers to, whereas the body is usually encoded using SDP [150]. Also similarly to SIP, RTSP follows the HTTP client-server paradigm in its operation, however it is a stateful protocol, due to the need to offer operations like "fast-forward" and "rewind" within a stream. In addition, RTSP is independent of transport layer protocols and can work equally well over TCP or UDP.

3.3.5 SIGTRAN

In certain VoIP-GSTN interworking scenarios, such as the MGC-SG communication within a decomposed gateway, it is necessary to extract high-level GSTN signalling information and deliver it to another component over IP. This process is more complex than a simple parsing of GSTN signalling messages and encapsulation of their data in IP datagrams, since the performance and reliability levels of circuit-switched networks are not easily met by conventional TCP/IP or UDP/IP networks. To address this issue, the IETF has developed a number of cooperating standards under the SIGTRAN framework [282].

In SIGTRAN connections, GSTN signalling information (e.g., SS7 ISUP messages [200]) is carried over IP, instead of using the native GSTN transport protocol or mechanism (e.g., MTP [199] in the case of SS7). For this purpose, a new transport layer mechanism, the Stream Control Transmission Protocol (SCTP) [355], has been specified. SCTP is a connection-oriented protocol that offers reliability over IP, without incurring the performance penalties inherent in TCP. It runs directly on top of IP and below an adaptation layer specific to the signalling protocol being transported (e.g., ISUP). This way, the signalling protocol can operate transparently over IP, without realising that its transport mechanism has changed.

3.3.6 MGCP and H.248/MEGACO

Within a decomposed VoIP gateway, communication of the MGC with the MG(s) is implemented in a master-slave fashion. This concept can be realised locally by Proprietary Device Control (PDC) techniques, such as interprocess communication (IPC), in the form of local or remote procedure calls across applications. A similar, albeit more standardised approach, is represented by the Message Bus (MBUS) [284], a signalling mechanism for co-ordinating co-located conferencing applications (e.g., a number of MBONE tools running on a LAN) that bypasses PDC limitations. However, both methods are localised and thus not suitable for location-independent decomposition. H.323 has also been examined as a solution, but eventually the need for a specialised protocol became evident [146].

Two standards have been proposed for spacing afar the MGC from the MGs, the Media Gateway Control Protocol (MGCP) [5] and MEGACO (later on renamed as Gateway Control Protocol) [134], [136] also known as H.248 in ITU-T parlance [187]. MGCP was derived from a combination of two earlier protocols, IP Device Control (IPDC) and the Simple Gateway Control Protocol (SGCP) [128]; it predates H.248/MEGACO and, although not intended to be the prevailing specification, it was promoted by vendors who wanted a standardised way for Media Gateway control, without waiting for H.248/MEGACO to be finalised.

Both MG control protocols work by abstracting the media interfaces of an MG (called *endpoints* in MGCP and *terminations* in H.248/MEGACO) and their various associations (called *connections* in MGCP and *contexts* in H.248/MEGACO). They also offer MGCs a powerful set of commands for remotely managing MG resources (e.g., logically joining interfaces to create peer-to-peer or conferencing calls through an MG), handling events (e.g., dialled digits) and reporting call statistics (e.g., call duration or amount of traffic carried).

Media Gateway architectures implementing either of these protocols can vary from single, integrated devices, to loose networks of components, each contributing its functionality to the overall capabilities of the (distributed) gateway. MGCP and H.248/MEGACO can function equally well for small and large-scale networks and run over a variety of network technologies, including IP [297], Frame Relay [130] and ATM [258]. H.248/MEGACO, however, as newer and more worked upon, offers a richer feature set, greater flexibility, improved security and the simultaneous backing of the ITU-T and the IETF; therefore, it is more likely to become the universal MG control standard in the medium to long term.

3.3.7 The Softswitch Model

The decomposed gateway model and the standardisation of the MGC-MG interface, combined with the constant increase of processing power found in off-the-shelf hardware, has made possible the creation of advanced telephony switches, the functionality of which is mostly implemented in software, running on high performance dedicated computers that are equipped with appropriate physical interfaces (e.g., line cards). These devices are called *softswitches* and they are increasingly seen as replacements to the monolithic, complex and expensive Class 4 and Class 5 switches traditionally used in the GSTN (although, strictly speaking, they differ in that they do not handle voice) [253], [281], [391], [394].

A softswitch is essentially a Media Gateway Controller with multiprotocol signalling support and possibly additional control functionality, capable of interfacing VoIP and GSTN networks, as well as supporting thousands or tens of thousands of calls via one or more associated Media Gateways. Softswitches can achieve similar reliability levels ("five nines") to GSTN switches, they are comparably scalable and they also offer the same set of functions (including signalling interworking, admission control, call management, Intelligent Network services, call routing, accounting and billing) [281]. Multiple softswitches can be chained together via the SIP-T protocol [378], as shown in Figure 3.7.

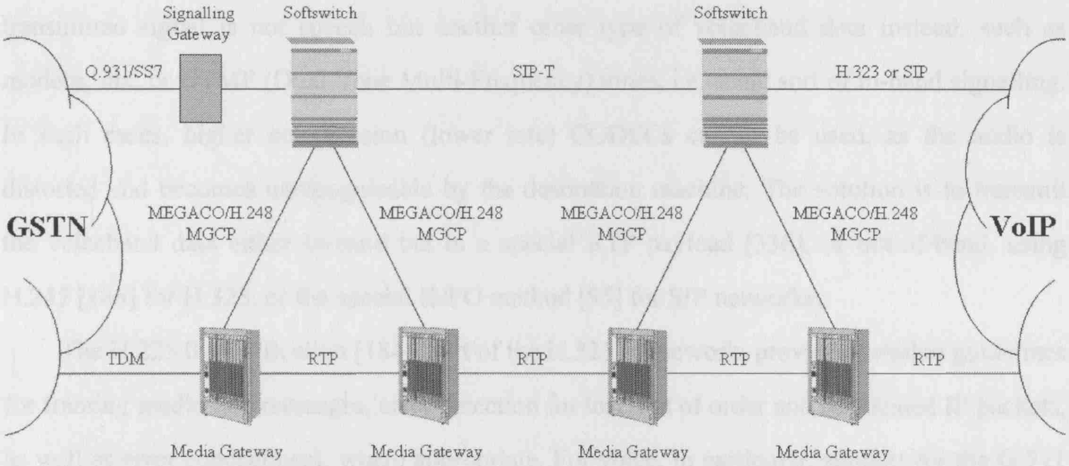


Figure 3.7: The softswitch architecture

The importance of the softswitch concept is threefold [391]. First, it removes the need for specialised and expensive switching hardware, thus reducing costs and allowing many more companies to compete with traditional telecommunication companies (telcos), producing far cheaper call charges. Secondly, it decouples the signalling and the media paths, allowing much more scalable implementations. Third, it facilitates the development and deployment of new services, due to the high programmability of its nature. As a result, softswitch architectures are increasingly found in commercial VoIP networks and are considered fundamental in deploying next generation scalable telephony applications [281].

3.4 The User Plane

The User Plane integrates numerous media processing operations, ranging from encoding/decoding to post-reception error control. The Real-time Transport Protocol (RTP) [337] is the universal standard for media encapsulation.

3.4.1 Media Processing

Multimedia traffic in IP networks is digitised and encoded in the appropriate format at the sender and delivered to the receiver using RTP packets, where it is decoded and converted back to its natural (analogue) form. Many CODEC standards exist for the various media types (e.g., audio, video) and the most common are (or can be) supported by both H.323 and SIP implementations. For VoIP, in particular, special measures need to be taken when the transmitted signal is not speech but another other type of voiceband data instead, such as modem, fax, or DTMF (Dual-Tone Multi-Frequency) tones, i.e. some sort of in-band signalling. In such cases, higher compression (lower rate) CODECs cannot be used, as the audio is distorted and becomes unrecognisable by the destination machine. The solution is to transmit the voiceband data either in-band but in a special RTP payload [336], or out-of-band, using H.245 [186] for H.323, or the special INFO method [95] for SIP networks.

The H.225.0 specification [184], part of the H.323 framework, provides detailed guidelines for framing media into messages, error detection for lost, out of order and duplicated IP packets, as well as error concealment, where appropriate. For voice, in particular, support for the G.711 CODEC [181] is mandatory [188]. In the case of data, encoding, encapsulation and transmission are handled by accompanying protocols of the T series, such as the T.38 [205] for fax over IP and the T.120 [206] for *whiteboard*-style conference applications.

SIP, on the other hand, does not mandate support for any particular CODEC, voice or other [318]; instead, the functionality described in H.225.0 is incorporated in accompanying IETF standards and related papers [287], [288], [303]. Many SIP implementations (particularly softphones), however, tend to support modern audio CODECs such as iLBC [3] or Speex [U50], in addition to more traditional ones, like G.723.1 [182].

3.4.2 The Real-time Transport Protocol (RTP)

RTP [337] is an end-to-end, application-layer protocol that provides a mechanism for transmission of real-time media like audio and video across a network, where the term "real-time" implies a stream-based transmission for which the original framing and timing of data is preserved and can be accurately reproduced at the receiving workstations, courtesy of the applications implementing the protocol. The core specification actually defines a pair of protocols, RTP itself for the media encapsulation (user plane) and the RTP Control Protocol, RTCP (control plane), which acts as a quality feedback mechanism, conveying information like packets and bytes transmitted or lost, delay, delay variation (jitter) and participant information such as names and e-mail addresses, for every RTP-encapsulated media session.

User-generated multimedia streams (VoIP or other) are encoded by the appropriate CODEC and then encapsulated by RTP, following the Application Layer Framing (ALF), End-to-End and Integrated Layer Processing principles [286], [337]. This, among other things, means that the protocol can be equally positioned either at the lower half of the application layer, or at the upper half of the transport layer, in the TCP/IP stack. In both cases, for interactive sessions run over IP the preferred transport protocol is the (unreliable, but time-efficient) UDP, due to the delays inherent in TCP operation (e.g., because of slow start or retransmissions), although TCP can be used instead for one-way transmissions, such as in streaming [128], [334]. This preference mandates that RTP enhances UDP with functionality necessary to overcome the lack of transport reliability and synchronisation capabilities.

More specifically, RTP offers:

- **Reliability**, via the *Sequence Number* field of its header, which allows the discovery of packet loss, sequencing and duplication problems caused by UDP/IP. Additional information useful for this purpose is provided by RTCP report packets, which give feedback to the transmitter and the receiver of a stream, about network conditions such as congestion and delay variation (jitter).
- **Synchronisation of media streams.** To assist applications in maintaining timing dependencies and to synchronise non-periodically transmitted media streams, RTP includes in its packet header a *Timestamp* field, derived from a linear monotonical clock, which usually increments at a higher rate than the smallest block size of the stream (e.g., audio sample rate).
- **Support for multiple payload types.** RTP can act as a packet-based delivery mechanism for a wide variety of media. Each RTP packet carries samples of a particular payload, i.e. of a single medium, encoded in a specific format (e.g., G.711 A-law speech [181]). The *Payload Type* field of the packet header provides unique identification of media encodings and appropriate specifications (called *profiles*) are used to map a set of payload type to their corresponding media formats. Most common payload types are mapped to static (pre-assigned) values, whereas a range of dynamic payloads is also available, to cope for the development of new schemes, without breaking existing applications.

- **Packet source tracing**. The particular source (e.g., a single audio microphone or video camera) of each RTP stream is uniquely identified in a session by virtue of another header field, the *SSRC* (Synchronisation Source), which is independent of the network address used and is capable of supporting tracing of a packet's origin even in cases when an intermediate device (such as a mixer or a network gateway) is included in the transmission path.
- **Versatility.** RTP does not assume the existence of any particular underlying protocol, only requiring that the latter provides framing and is not affected by any addressing changes in the network layer. In effect, many different lower layer protocols offering end-to-end connectivity can be used for RTP. For instance, H.323 Annex C [188] specifies RTP transmission directly over ATM AAL5 [193], bypassing UDP and IP.
- **Lightweight operation.** RTP follows the end-to-end argument [324] and thus works as an informational mechanism, relaying media and media metadata that can be exploited by the implementing applications in a way they seem fit. Operations like stream multicasting, multiplexing, mixing, synchronisation, compression, flow control, resource reservation, error concealment and encryption are delegated to the communicating endpoints and the protocol itself does not guarantee timely delivery, quality of service, or error-free transmission.

After the appearance of RTP Version 1 in the visual audio tool (vat) in 1991 [257] and the standardisation of the current Version 2 in RFC 1889, in 1996 [286], the protocol has been more or less stabilised. Its current form, published in 2003, added only minor improvements to certain rules and algorithms while maintaining Version 2 numbering [337], thus further justifying the maturity hypothesis. As a result, RTP remains the universally accepted standard for encapsulation and transmission of real-time multimedia over IP networks, either in interactive communication (usually initiated by H.323 or SIP), or for streaming (typically set up by RTSP).

3.5 Call Routing

VoIP call routing is both an application layer and a network layer process. At the application layer, signalling servers and similar devices act as call routers that relay control messages among communicating terminals, which, during a call, usually exchange media "directly" (i.e., without the intervention of any other application layer media device), save from interworking or conferencing functions. At the network layer, IP implements global communication via its long-established and field-proven routing protocols [144], [167], in essence offering a full mesh that interconnects all components of the VoIP network.

3.5.1 Signalling and Media Paths

Following the GSTN example, the application layer signalling and media paths in IP Telephony networks are separate, and can differ significantly in length and variation. This approach facilitates scalability, reliability and service provision, but it is not free of implications [295].

Signalling protocols, more specifically, route calls on a hop-by-hop basis, over a single path defined by an initial setup message that all subsequent messages also follow on either direction, apart from intentional path changes (e.g., mid-call rerouting) or extraordinary circumstances (e.g., a link failure). The number of application layer signalling hops is usually kept small, and that is expected even from SIP which, as discussed, follows the longer-path SMTP routing model [227]. However, with the exception of direct signalling, in larger networks it is quite conceivable that signalling messages for long distance calls will have to traverse a number of providers from source to destination, thus increasing the number of hops. The operation of IP at the network layer can complicate the situation further.

The media path, by contrast, assumes "direct" (zero-hop) application layer connectivity inside the IPTN, where the endpoints are either two VoIP terminals (for pure IP calls), or a VoIP terminal and the gateway that can reach the destination phone (if an SCN is involved). This kind of connectivity is only susceptible to the fluctuations introduced by IP routing at the network layer; however, the need for multiple intermediary devices (e.g., transcoding gateways, MCUs for centralised conferencing, media mixers, or even protocol translators like IPv4-IPv6 converters) cannot be ruled out as IP Telephony networks grow in size, expand to offer interoperability with the GSTN and other SCNs, or comply with security policies by tunnelling their traffic through firewalls, Session Border Controllers or even lawful interception devices [295]. In such cases, the media path may have to be segmented by these intermediaries, and setup be performed on a sub-optimal, segment-by-segment basis; alternatively, a unified end-to-end mechanism (e.g., source routing) for the IP part of the network may prove to be more efficient. Until this happens, however, the establishment of the media path at the application layer will continue to be considered simpler and thus the signalling process will remain the main challenge of IP Telephony call routing.

3.5.2 Translation Functions

Due to the prevalence of the GSTN usage model, IP Telephony applications often need to perform address translation, in order to convert the human-friendly address alias of the destination (e.g., a telephone number or an e-mail address) to an IP address understandable by the network. Depending on implementation, this translation function can affect the construction of the signalling path at the edge of the VoIP network (e.g., whenever different directory servers need to be consulted), although the problem is usually delegated to the signalling server to which the calling terminal is associated. However, there exist application layer translation mechanisms (e.g., Application Layer Gateways acting as firewalls) that can locally complicate this process.

More difficult is the case with address and/or other translation functions taking place at the transport and network layers of the TCP/IP stack. A typical example of such a function is the conversion between GSTN addresses (E.164-based telephone numbers) and VoIP addresses (e.g., H.323 aliases or SIP URIs). The IETF has developed the ENUM framework [109] which specifies how DNS [273] can be used for storing E.164 identifiers [172] and discovering the services associated with them, so that telephone numbers can be used as a universal addressing scheme for all types of communication [260]. ENUM mappings have been specified for both H.323 [243] and SIP [291], and the ITU has also added appropriate supplements to the E.164 standard [172].

Another type of address translation, used by many organisations for IP address space preservation and/or security reasons, is Network Address Translation (NAT), a technique that modifies the IP address and/or TCP or UDP port associated with an IP Telephony signalling or media path, thus requiring special techniques like TURN [311], STUN [319], ICE [312] and RSIP [39], [40] for connectivity to be maintained [128], [159]. As discussed in the security section, firewalls and similar devices also need modifications to work with VoIP. To address the most common of such connectivity scenarios in a unified and secure way, the IETF is developing specific standards for intermediate device ("middlebox") communication (MIDCOM) [351], [362], whereas numerous proprietary mechanisms like Microsoft's Universal Plug and Play (UPnP) [266] and Ridgeway Systems' IPFreedom have been proposed by commercial vendors [67], [360].

The most complicated aspect of the call routing procedure, of course, takes place at the backbone, particularly for communication scenarios that involve heterogeneous network paths.

3.5.3 Gateway Location

In hybrid communication scenarios (i.e., those involving heterogeneous networks), the complexity of each of the different network clouds, their mutual incompatibility and the large differences in provider policies, mandate that call routing be conducted individually, as no unified, end-to-end mechanism can be proposed. In IP Telephony communication, more specifically, whenever a segment of a call resides on the GSTN, the selection of an appropriate (according to some criteria) gateway for the necessary translation of signalling and media traffic becomes crucial [315], [316]. In calls from a GSTN terminal to an IP Telephony terminal, this process can be reduced to a simple directory lookup based on the destination alias, since the telephony switch from which the call will exit the GSTN will be collocated with (or otherwise know of) the proper gateway, and a solution could be having a special arrangement (e.g., a separate country code) for the IP Telephony address space [295]; on the opposite direction, however, VoIP-to-GSTN calls will be more complicated, due to the possible existence of numerous alternative routes to enter the GSTN, as a result of topology, policy, feature and user requirement differences across the two networks [316]. Therefore, identifying the most suitable route will depend on gateway capabilities (such as pricing, supported signalling protocols, selection of CODECs, range of reachable phone numbers, total circuit capacity, available circuits and desired QoS) at the particular time of a call [315].

The capabilities of each gateway can be propagated to the rest of the VoIP network in two ways: either *manually*, in which case the signalling servers directly connected to the (MGC of the) gateway are configured with this information (and so is, of course, the Media Gateway), or *automatically*, via a special-purpose protocol. The manual configuration case presents a number of obvious disadvantages:

- Poor scalability, since the routing tables have to be updated by hand.
- Inconsistency, since both the signalling server and the gateway must store the same type of information and update it simultaneously.
- Lack of dynamic network adaptability, since changes to the situation of the gateway (e.g., due to overload) cannot be automatically reflected to the signalling server.
- Dependence on third-party protocols (e.g., DNS in the case of SIP) for discovering the next signalling server (next hop) in the path from source to destination.

Automatic configuration can solve these problems and a number of generic techniques have been suggested for this purpose, including DNS [273], LDAP [384], SAP [152], SLP [141], Multiprotocol BGP [23], web-based indexing systems like Harvest [41] and multicast advertisements [315]. However, a specially developed IETF protocol, TRIP, has emerged as the preferred solution.

3.5.4 The Telephony Routing over IP (TRIP) Protocol

TRIP [314] is largely based on BGP-4 [306] and used for distributing gateway-related call routing information necessary for completing a call that originates in a VoIP network and terminates in a GSTN network. The protocol defines a Location Server (LS) entity, which stores and distributes to other LSs information about gateways and the phone numbers reachable through them, along with certain route metrics (e.g., MG capacity or call price). A set of gateways and LSs managed by a single administrative authority is called an IP Telephony Administrative Domain (ITAD) in TRIP parlance.

Like BGP-4, there are two versions of TRIP: *Internal* (I-TRIP), which is used in an intradomain fashion for propagating detailed telephony routes within an ITAD, and *External* (E-TRIP), which disseminates summarised routing information across ITADs. For synchronising LS databases, I-TRIP employs a flooding mechanism based on the ones implemented in OSPF [275] and SCSP [251], whereas E-TRIP uses peer-to-peer connections between LSs positioned at ITAD borders. In addition to intra-domain synchronisation, peer communication and the inter-domain transport mechanism, TRIP borrows several other concepts from BGP-4, including the finite state machine, message formats and attribute definitions [314]. Contrary to BGP-4, however, TRIP is an application layer protocol, it exchanges phone number prefixes (instead of IP address prefixes), it can work even with partial connectivity across routing domains (ITADs), it does not require full mesh connectivity among LSs and, crucially, it is not involved in packet forwarding [316]: LSs only act as databases for dynamically collecting gateway-related information and leave the task of forwarding control packets to the signalling servers at the application layer and to IP routers at the network layer.

TRIP supports both H.323 and SIP, and can handle additional signalling protocols, by design. Each route propagated by an LS contains the IP address of a signalling server (e.g., a proxy server, a gatekeeper, or an MGC) controlling an MG that can reach certain telephone number prefixes. When a signalling server receives a call setup request for a phone number, it contacts a TRIP location server it somehow knows of, gets a list of possible routes to gateways (i.e., MGCs or other signalling servers controlling MGs) that can reach this phone number, and forwards the request accordingly.

Location servers learn about gateways and their attributes via the Telephony Gateway REgistration Protocol (TGREP) [17], which is a modified subset of TRIP used to convey dynamic gateway resource information. A TGREP client runs at each gateway (at the MG and perhaps the MGC), collects its attributes internally and passes them on to the known LSs. This network model further assumes that:

- Each signalling server communicates with one LS (via a function call, or by another unspecified mechanism).
- The signalling server and its corresponding LS are usually co-located, in the absence of any protocol for their communication.
- Each signalling server may route calls to more than one gateway.
- Each gateway may talk to more than one LSs (via TGREP).

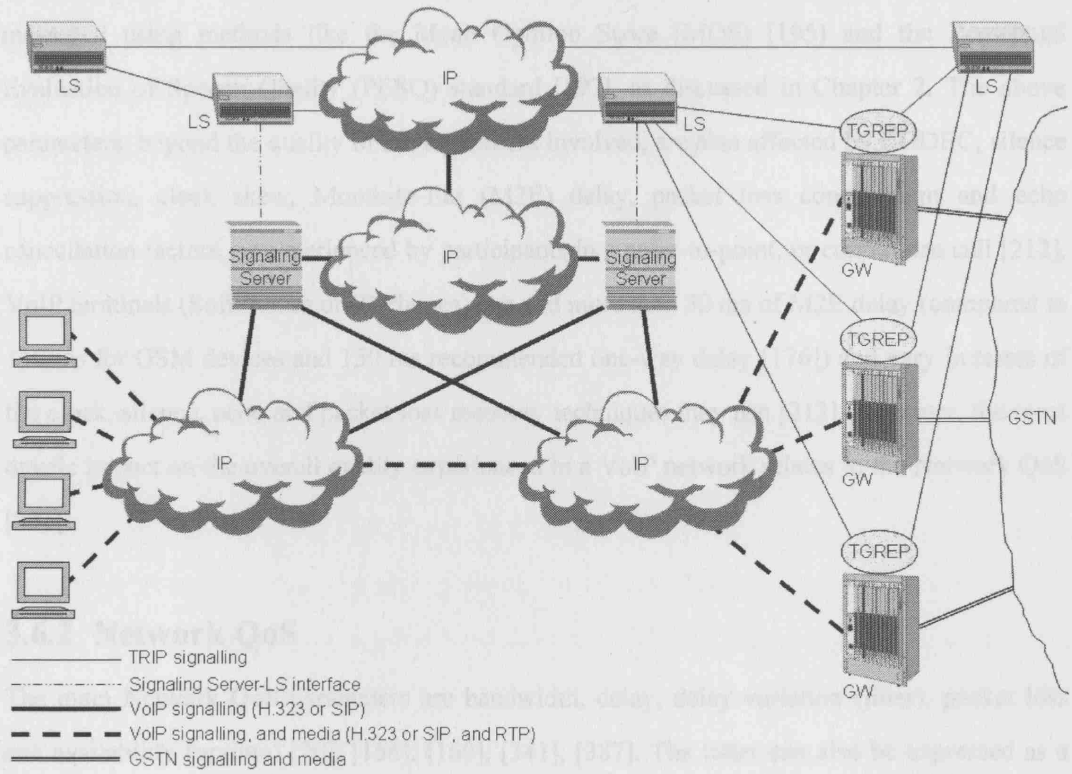


Figure 3.8: A TRIP-enabled VoIP network

An example of this architecture is shown in Figure 3.8. Generally speaking, TRIP can be classified as a service location mechanism, having obvious functional similarities to protocols like SLP [141], although it is customised for IP Telephony gateway discovery.

3.6 Quality of Service (QoS)

The overall quality of service offered by a VoIP network is a composition of the voice quality perceived by end users through the client applications that interface them to the network ("User QoS"), and the behaviour of the network itself ("Network QoS"), as discussed in Chapter 2. Both types of quality can be quantified through specific performance parameters, which, within certain limits, correspond to acceptable levels of service.

3.6.1 User QoS

The main parameters of User QoS are identifiability, intelligibility, interactivity, fidelity and naturalness of the voice signal reproduced at the receiver [156], and their combined effect is measured using methods like the Mean Opinion Score (MOS) [195] and the Perceptual Evaluation of Speech Quality (PESQ) standard [197], as discussed in Chapter 2. The above parameters, beyond the quality of the equipment involved, are also affected by CODEC, silence suppression, clock skew, Mouth-to-Ear (M2E) delay, packet loss concealment and echo cancellation factors, as experienced by participants in a point-to-point, or conference call [212]. VoIP terminals (SoftPhones or IP Phones) can add more than 50 ms of M2E delay (compared to 110 ms for GSM devices and 150 ms recommended one-way delay [176]) and vary in terms of the clock, silence, echo and packet loss recovery techniques they use [212]. However, the most drastic impact on the overall quality experienced in a VoIP network relates to the Network QoS [231].

3.6.2 Network QoS

The main Network QoS parameters are bandwidth, delay, delay variation (jitter), packet loss and availability (uptime) [70], [156], [160], [341], [387]. The latter can also be expressed as a combination of one or more values of the former (e.g., an unavailable system can be considered as having zero bandwidth, 100% packet loss, or infinite delay), whereas it has been observed that delay, delay variation and packet loss are often interrelated and come in bursts [212], [213], [214]. Additional network effects, such as packet reordering, can also be considered with relation to Network QoS, however their impact (and the recovery from it) can be handled relatively easily in terms of time and processing effort, so as to be neglected [231].

IP Network QoS improvement techniques can be broadly classified into three categories (as per Chapter 2): *policy-based*, where call admission control and/or higher bandwidth (overprovisioning) are employed; *tuning*, used for improving network performance in specific cases, such as slow links or aggregated stream traffic; and *signalling*, which is the most complicated but also most effective one, where the main methods are Integrated Services, Differentiated Services and IP Switching [244], [388].

Integrated Services (IntServ) [42] aims at improving QoS in IP networks so that they can support all types of traffic, including real-time, over a common infrastructure. This is proposed to happen by admission control and reservation of resources on an end-to-end, per flow (media stream) basis, via a signalling mechanism, the Resource ReSerVation Protocol (RSVP) [43]. RSVP is used by senders to notify (via a special PATH message) potential receivers about the characteristics of the traffic that will be transmitted; receivers that want to get this traffic, reply with a reservation request (sent in a special RESV message) which follows the same path as the initial notification and, if accepted by all routers (and, eventually, the sender itself), transmission at the specified QoS level can start. Two QoS levels are supported by RSVP: Guaranteed Service offers hard QoS performance (i.e., guarantees parameters like bandwidth and packet loss); Controlled Load, on the other hand, is less demanding, as it simply makes the reservation path behave as if it were on an uncongested network, which is good enough, however, for most communication scenarios [128]. RSVP is the only mechanism for offering ATM-style, virtual circuit QoS over IP networks, however it is a complex protocol that does not scale well, due to its per-flow operation [341].

Differentiated Services (DiffServ) [34], named as to indicate an opposite approach to Integrated Services, accept the high difficulty of offering hard QoS guarantees over IP and instead opt for a more conservative approach, in which multiple traffic flows are grouped together into classes that can receive better treatment by network routers, according to their agreed level of service. Traffic classification happens on a per-packet basis, by reusing the Type of Service (TOS) byte of the IP header [297] as a Differentiated Services Codepoint, which is associated with a specific Per-Hop Behaviour (PHB), identifying its treatment by the network. DiffServ does not implement admission control and relies on common routing protocols for traffic forwarding (i.e., it does not establish any special paths), which allows much simpler and more scalable operation compared to IntServ, at the expense of achieving only crude performance improvements, compared to the fully guaranteed QoS possible with RSVP.

An alternative (but not incompatible) approach to both IntServ and DiffServ is IP Switching, standardised around the Multi-Protocol Label Switching (MPLS) specification [310]. MPLS removes most of layer 3 (IP header) processing and instead uses a short label to rapidly switch packets over a path of special Label Switching Routers (LSRs), called Label Switched Path (LSP), and established before communication starts via a signalling mechanism, such as the Label Distribution Protocol (LDP) [4]. The particular LSP each packet follows depends on its Forwarding Equivalence Class (FEC), i.e. the class of service offered to the traffic flow in which the packet belongs. This "layer 2.5", ATM-like operation allows MPLS networks to offer very high speeds and improved QoS, compared to normal IP networks.

The three approaches to IP Network QoS are not incompatible. In certain scenarios, IntServ, DiffServ and MPLS can be mutually combined and also cooperate with other LAN and WAN protocols [29], [112], [239], thus complementing the role of each other. However, the difficulty of applying QoS on a best effort technology, like IP, remains, and, as already discussed, the techniques used to solve it almost invariably end up imposing some sort of connection-oriented operation over the inherently connectionless service implemented by the Internet Protocol.

3.6.3 Acceptable QoS

While target (user and network) QoS levels remain at the theoretical maximum offered by traditional GSTN operators (e.g., MOS of 4 or more, "five nines" availability, high level of customer services) [26], it should be noted that users can tolerate considerably lower QoS levels, in exchange for lower prices and/or higher flexibility. This has been the case since the early 1990s, with the deregulation of the fixed line market, which created several non-incumbent operators that tend to cut costs by reducing service levels (e.g., the "five nines" guarantee, corresponding to around 5 minutes downtime per year) [70], [82], and the appearance of mobile telephony, which, by design, is not as reliable or high-quality (e.g., in terms of speech MOS) as landline telephony [253]. This paradigm shift in user expectations can accelerate, to a considerable extent, the adoption of the (often lower QoS) VoIP services.

3.6.4 Measuring QoS

The idiosyncrasies of the Internet Protocol and the rapidly increasing popularity of VoIP, create the need for assessing the QoS capabilities of an IPTN before its deployment. Several tools exist that automate this process for both user and network QoS. In the first case, the assessment is based on subjective or objective measurements producing MOS or similar scores, according to one of the generic methods (e.g., PESQ, which is particularly suitable for VoIP due to its resilience in varying network conditions, or the E-Model) presented in detail in Chapter 2, although their effectiveness on VoIP networks can vary [78], [100], [104], [156]. For network QoS, in addition to parameters like those described in E.437 [174] that are applicable to any packet technology, numerous IP-specific techniques have been proposed as well. Commercial applications, for instance, use built-in TCP/IP tools (*ping* and *traceroute*) [353], DNS [273] or SNMP [59] in order to built a layer 3 topology, on which simulations or real-time tests (e.g., by injecting certain traffic patterns) can be conducted [25], [U3]. Delay and delay variation emerge as predominant factors for QoS degradation (including, to an extent, packet loss) from many of the studies on Internet performance, pointing to similar results for individual IPTNs [36], [154], [213], [218], [288], [320]. Interestingly, even the (usually overprovisioned) network backbones can exhibit sub-optimal QoS when it comes to carrying voice traffic [142], [254].

3.7 Security

The main security components of a communications system are usually identified as *confidentiality* (privacy), *integrity* (data integrity, i.e. preservation of the actual information, and origin integrity, i.e. authenticity) and *availability* (ability to use the system as designed), while a further one, *non-repudiation* (transaction accountability) can be expressed as a combination of data and origin integrity [32], [33], [327], [342].

3.7.1 Threats

Generally speaking, security threats encountered on communication systems can be categorised in four main classes [33]:

- *Interception.* The attacker reads information exchanged between the communicating endpoints.
- *Modification.* Information exchanged is modified (changed or deleted).
- *Masquerading.* The attacker appears as somebody else, or gets access to the system by assuming the identity of a legitimate user, a situation also known as "identity theft".
- *Denial of Service.* The network becomes partially or fully unavailable to its legitimate users, as a result of an attack.

IP Telephony networks, in particular, as hybrids of both voice and data networks, inherit security risks from both of these networking worlds [343]. In that sense, VoIP systems remain vulnerable to both technical and non-technical ("low technology" or "social engineering") attacks [342], which are exacerbated by the added intelligence of digital, packet-based operation. Further threats arise due to the technology's own design vulnerabilities (e.g., protocol imperfections), and the software-based implementation of the majority of its functionality, which makes VoIP open to a wide variety of operating system and application attacks.

For circuit switching voice networks, common security threats include, among others [30], [271], [272]:

- Unsolicited calls. These are the result of inadvertent revealing or guessing (e.g., by some automated means) of a phone number and placing calls to it (e.g., for marketing purposes).
- Caller ID masquerading. The calling party blocks or alters the Call Line Identification (CLI) information normally transmitted by the network, so that the callee cannot know the origin of the call before answering.
- Account hijacking (identity theft). An attacker breaks into a legitimate telephone account (e.g., that of a normal customer, or a special account used for network testing purposes) and can place phone calls charged to it, or conduct other unauthorised activities using the assumed identity of the account's owner.
- Network intrusion. This includes a number of different attacks which result in capturing information carried over, or related to a telephone network. Eavesdropping of conversations is a classic example of such a threat.
- Denial of Service attacks. The operation of the telephone service is disrupted by unauthorised changing of calling plans, rerouting of calls, or disabling of the core functionality of local exchanges or other switches.

In data networks, security threats often include [32], [33], [342]:

- Sniffing/snooping. Passive wiretapping (interception) of information exchanged over a data communications network.
- Man-in-the-middle attacks. Active wiretapping of information, in which the attacker captures and selectively modifies exchanged data in order to conceal his presence, while simultaneously breaching the security of the network.
- Spoofing. The attacker poses as, or assumes the identity of, a legitimate user. This is a typical example of masquerading for data networks.
- Password cracking. Guessing user passwords, usually by automated means, such as special

software written for this purpose, with the intention of committing identity theft.

- Malware. A specific class of software applications (viruses, worms, trojans, backdoors), the execution of which results in malicious effects on individual machines or networks.
- Denial of Service Attacks. The attacker uses protocol imperfections, implementation bugs, or brute force in order to render a network inaccessible to its legitimate users.

In addition to voice-specific and data-specific security vulnerabilities, there are also potential combined threats, i.e. those applicable exclusively to IP Telephony networks [384]. A typical DoS attack in a VoIP network, for instance, could consist of artificially inserting delay and packet loss, resulting in the progressive degradation of user-level QoS and, eventually, the disruption of communication; executing a VoIP Trojan can lead to microphone hijacking, enabling the attacker to remotely listen to a user's private conversations; the recordings from eavesdropping on a conversation can be processed off-line, using tools like VOMIT [U63]; and a global directory of VoIP addresses, such as the one planned in the ENUM specification [109], opens the door to easy unsolicited communication (spam) in a worldwide scale, a threat already referred to as "SPIT" (Spam over Internet Telephony) [305].

A less often encountered, but equally serious, class of security threats stems from natural or artificial causes (e.g., weather disasters or terrorist attacks). To avoid (or, at least, cope) with such situations, users must have direct access to state-controlled services (e.g., the "911" emergency number in the USA and the "enhanced 911" specification for caller location in wired, wireless and VoIP networks [51], [52]) and, vice-versa, the state must be able to prevent the abuse of security mechanisms for illegal purposes (e.g., using strong encryption for impregnable communication among terrorist groups); hence, there is a case for emergency services support and for lawful interception (surveillance) of traffic in VoIP networks [105], [327]. Other forms of IP Telephony signalling and traffic interception may also be desirable by governments and/or organisations. Large telcos, for instance, often want to be able to identify VoIP calls flowing over their networks, and classify them as lower priority (or even block them entirely), in order to protect their GSTN-generated revenue stream [71]. This is a non-trivial process, since IP Telephony packets are difficult to distinguish from normal data packets flowing over the same network.

While VoIP, as a communications technology, faces numerous security issues, as discussed above, it is questionable whether it is often significantly less secure than conventional telephony. This is so because classic threats like eavesdropping require, at the very least, access to the network for installing special software (sniffers), knowledge of the actual path that signalling and media packets will follow, and a compatible CODEC to listen to the conversation, whereas the difficulty of an attack is raised exponentially if encryption or other security measures are implemented [159], [164], [305] (lawful interception [105], for example, is complicated partially due to these same reasons). That being said, the set of potential security threats is large, and therefore developing effective solutions is a necessity for contemporary and future networks.

3.7.2 Solutions

Securing IP Telephony networks is necessarily a complicated, heuristic and imperfect process, due to the wide variety and sophistication of potential threats. In most cases, typical data security tools (e.g., firewalls) and techniques (e.g., encryption) need modifications to work with VoIP and comply to its performance requirements [384], thus creating the need for an individual "VoIP Security" field in the world of data communications.

Generic security mechanisms for telephone and IP networks are also applicable in VoIP systems, with emphasis on confidentiality, integrity and authentication at the LAN (client), MAN (network access) and WAN (backbone) level [27], [32], [272], [326], [327], [342]. In addition to encryption, intranets and other corporate networks implement sophisticated multi-level access controls and information hiding mechanisms, via dedicated firewall [305], Network Address Translation (NAT) [39], [40], [311], [312], [319], [351], [362] and Session Border Controller (SBC) [158] functions. These can be implemented individually or as a combination (e.g., a firewall as part of an SBC), in devices that process or filter signalling and media packets in order to conceal internal phone numbers, IP addresses and topological details, so that direct attacks against individual machines (or groups of machines) in the network cannot be performed. Furthermore, secure coding guidelines can be applied so that IP Telephony operating system and application software vulnerabilities are minimised, depending on implementation [131], [339].

In addition to generic security mechanisms, there are also numerous protocol-specific ones, which can be encountered at all layers above the physical layer.

- **Data Link Layer**: This is the lowest layer at which a Virtual Private Network (VPN) can be implemented. The standard security mechanism for this purpose is the Layer 2 Tunnelling Protocol (L2TP) [370], which is a successor to the Point-to-Point Tunnelling Protocol (PPTP) [149] and the Layer 2 Forwarding protocol (L2F) [376]. L2TP extends PPP [344] for tunnelling network traffic over IP networks. As IP Telephony networks tend to be agnostic to layers 1 and 2, higher layer solutions are more suitable for securing VoIP applications.

- **Network Layer**: The secure version of the Internet Protocol, IPSec [222], is a collection of protocols and mechanisms that offers confidentiality, integrity and authentication services for IP networks, transparently to higher layer protocols. IPSec works either in transport mode, in which only the message payload in encrypted, or in tunnel mode, in which both the header and the payload are encrypted. Because it implements security without the need of modifying existing applications, IPSec is the preferred method for implementing VPNs, however VoIP applications may need special techniques to overcome the unavoidable performance penalties introduced by its operation [20].
- **Transport Layer**: The proposed standard for secure layer 4 communications is the Transport Layer Security (TLS) protocol [93], which offers confidentiality, integrity and authentication mechanisms, thus replacing the Secure Sockets Layer (SSL) protocol originally used for this purpose, particularly in HTTP transactions [110]. The connection-oriented nature of TLS, as well as the modification of the standard socket API to add security mechanisms, make it less attractive for VoIP applications, particularly interactive ones, running over UDP.
- **Application Layer**: All VoIP-specific protocols reside on this layer, and most have special provisions for security. At the control plane, H.323 security mechanisms are described in H.235 [185], which specifies a wide range of protection mechanisms, including confidentiality, integrity and authentication services. SIP, on the other hand, as a more loose framework, relies on a combination of techniques to achieve the same level of protection; in addition to the core mechanisms [318] for protecting the header section of each message and securing communication end-to-middle [283] or end-to-end [8], there is also the option of encrypting the message body via SDP [150], [159] to avoid disclosure of information related to media path setup, or using Secure MIME [304] for confidentiality, integrity and authentication [102], [290]. For hop-by-hop security in the backbone, SIP relies on transport and network layer mechanisms [323]. Security mechanisms have also been specified for other control plane protocols, including SIGTRAN [249], MGCP [5], H.248/MEGACO [136], [187] and TRIP [314]. For the user plane, the main application layer security mechanism is Secure RTP (SRTP) [24], which provides confidentiality, message authentication, and replay protection for both RTP and RTCP streams. Admission control mechanisms, either integrated with H.323 and SIP, or as measurement-based add-

ons that allow calls to be placed only if certain network conditions are met [121], [255], [374], can also be classified as application-layer security techniques.

Mechanisms also exist for offering emergency calling and traffic surveillance over VoIP networks. The former are harder to implement, due to the unreliable nature of IP networks and the additional fact that certain technical provisions (e.g., electrical current supply) are normally prerequisites for the operation of VoIP devices. However, by enhancing the resilience, call setup times and call handling performance of a VoIP system so that it can perform as requested in extraordinary cases, emergency services can be supported and have been specified for both H.323 [188] and SIP [270]. Lawful interception, on the other hand, is complicated compared to the GSTN, by potential path fluctuation in VoIP signalling and media traffic, the variety of protocols involved, as well as the strong security mechanisms that may be applied. The IETF has produced some relevant guidelines [16], [169] and numerous implementations for both H.323 [265] and SIP [350] exist, although a common standard has yet to be developed. Solutions also exist for intrusion detection [380] and for VoIP traffic blocking through proprietary mechanisms like the one proposed by Narus [U31].

In summary, it can be argued that, although not necessarily less secure than conventional SCNs (and particularly the GSTN), IP Telephony systems are vulnerable to both "phreaking" (breaking into telephone networks) and "hacking"/"cracking" (violating data networks), as well as special purpose attacks, which make their protection a particularly cumbersome process. This complexity leads to an inevitable trade-off with usability that can evolve into a security risk itself, as suggested measures and policies are not followed by end users [32], [33]; it can also be considered as one of the main reasons why VoIP Security needs to be addressed as a separate field, requiring significant amounts of additional research and development effort to reach the quality and efficiency levels of other areas of the technology [305].

3.8 VoIP Application Areas

Since its popularisation in the mid-1990s, Voice over IP has been expanding its reach into current and newly emerging realms of applications and user bases, starting from its role as an enhancement or replacement of traditional telephony services, and expanding into various forms of IP-based voice communication, as discussed in the following sections.

3.8.1 Telephony

Outside academic and other research facilities, Voice over IP applications were first exploited by end users as a cheap (free) replacement to local and long-distance telephony services. This "home" version of VoIP has become universally available through software bundled with operating systems (like NetMeeting in Microsoft Windows) and other popular applications, like Instant Messaging programs, in addition, of course, to softphones, IP Phones and small ("residential") gateways used to connect to the network. After 2000, many established or new companies (e.g., Vonage [U65]) also started offering commercial VoIP services for home and business customers in various countries. These IP Telephony Service Providers (ITSPs) can offer reduced per-minute calls charges by operating (or contracting) a number of gateways worldwide that allow them to exit to the GSTN locally at each destination and thus incur local call charges (further reduced by mass purchase of call minutes from the corresponding telcos), while transporting voice traffic over their own, inexpensive managed IP backbone. In addition to the ITSP service model, wireless voice solutions based on IP, including 3G Mobile [54] and voice over wireless LANs (VoWLAN) [122] are becoming very popular in both the home and the business sectors.

The needs of the business market, more specifically, are addressed by providers through a wide variety of products and solutions, following the tradition of the GSTN. Digital telephony Private Branch Exchange (PBX) vendors originally extended their circuit switching products to accommodate VoIP through special expansion "daughterboards" [385] (e.g., Nortel and its Meridian product series [U33]), but soon switched to fully-fledged "IP PBX" products. Data companies (like Cisco [U7]) have followed the same path, albeit starting from the addition of voice services to their routers (again, via expansion cards) and then switching to native IP PBXs. Several VoIP gateway products have also emerged, whereas software-only solutions for both IP PBXs and gateways exist, like the popular, Linux-based Asterisk platform [U1]. Furthermore, many traditional GSTN solutions have migrated to VoIP, resulting in the deployment of IP Call Centres and IP-based outsourced PBX services ("IP Centrex"). Not surprisingly, E-Commerce and wireless voice applications are also expanding their user base very rapidly [81], [159].

In addition to "Home VoIP" and "Business VoIP", the "Enterprise VoIP" category is also thriving. Medium to large companies and organisations are using IP Telephony on their corporate networks, either natively, or as a PBX interconnection solution, to join remote premises, thus bypassing expensive toll call trunks. Major carriers worldwide (including BT

[U2]) have almost universally switched to IP for their backbone networks [79] and technologies for offering end-to-end, transparent voice solutions through VoIP and GSTN "islands" of connectivity are being deployed, including large-scale softswitches translating Q.931/ISUP, SS7, H.323, SIP, MGCP and H.248/MEGACO signalling for Media Gateways that support thousands or tens of thousands of simultaneous calls [281], thus replacing monolithic (and expensive) GSTN Class 4 and Class 5 switches [26]. Like with the home and the business sectors, wireless voice solutions are also attractive for the enterprise level, particularly in the form of the IEEE 802.16 (WiMAX) standard [81], [280], capable of offering last-mile voice services to end users and organisations at high speeds and without the need for expensive cabling installations.

3.8.2 Peer-to-Peer VoIP

Peer-to-peer (P2P) networks are distributed systems without any centralised control or hierarchical organisation, in which nodes run applications equivalent in terms of functionality [246]. Their main characteristics are high *scalability* (large number of nodes can be accommodated) and *reliability* (no single point of failure exists), at the expense of increased communication latency and bandwidth scarcity. In such networks, information (e.g., a set of files) is stored in more than one node and must migrate when its original location becomes unavailable, while nodes are assumed to join and leave (or fail) frequently. A maintenance protocol runs in the background to ensure connectivity is achievable at all times.

P2P technologies have become widely known in the Internet due to communities built around protocols like Napster, KaZaA and Gnutella, used for sharing media content (mainly MP3 music and videos), often in violation of copyright regulations. However, a large amount of attention is also being placed on this area by academics, resulting in advanced protocols like CAN, Chord, Pastry and Tapestry [246]. These create connected routing overlays on top of IP and can map a key (such as an IP address, a URL or a file name) to a node in the system in logarithmic time - in other words, they can route information to a destination node identified by a key, over a logarithmic number of hops.

Although at a first glance peer-to-peer technologies defy the client-server or master-slave model encountered in IP Telephony signalling and can be too slow for media traffic, they are attractive, due to their scalability and reliability, at least for the first purpose. P2P extensions to SIP have been proposed and are also assumed applicable to H.323 [215], [346], whereas a new IP Telephony application, Skype [U48], based on the KaZaA P2P file sharing model, offers encrypted voice, video and instant messaging services using proprietary signalling and media transport schemes [21] but with sustainable high QoS and satisfactory reliability, thus attracting more than 100 million users in its first two and a half years of operation, around 5% of which are often simultaneously online. In fact, the Skype operational model is unique enough to be considered as an entire IP Telephony framework on its own, the application itself can be viewed as a platform for offering new communication services and its overall success in popularising VoIP can only be compared (in proportion) to the revolution caused by VocalTec's Internet Phone about a decade earlier. On another front, a proposal for a Location Service Protocol (LSP), implemented via a P2P version of SIP and intended for the communication between TRIP LSs and signalling servers, is under development in the IETF [215].

Overall, P2P technologies present an interesting field of exploration for future developments in IP Telephony, particularly for the control plane, for both home and business use.

3.8.3 New Services

The functionality of IP Telephony networks can be upgraded fairly easily, since intelligence is located at their edges (at least as far as the layer 3 operation is concerned), as opposed to the centrally operated GSTN, which requires a non-trivial reprogramming of backbone switches for the same purpose. Because of this convenience, the deployment of new applications, combining multimedia with conventional data and exploiting the intelligence of converged network architectures, is always feasible and can be considered one of the major driving factors for the future of VoIP [253], [331]. Following such an approach, the aim is not only to port the advanced applications of the ITU-T long-standardised "Intelligent Network" (IN) framework [120], [203] from the GSTN to IP, but also to deploy newer, packet-based services, either designed from scratch, or resulting from an innovative re-use of existing ones.

Popular categories of advanced services (partially) implemented using VoIP are: *conferencing*, where information such as voice, video, and whiteboard is interactively exchanged over IP [245]; *streaming*, where IP-based media on demand applications like IP TV are included [69]; *unified messaging*, where a single digital "inbox" is used as a common repository for e-mail, speech, fax and data messages, integrated with VoIP (e.g., via embedded call-back URLs) [295]; *gaming*, where online communities of users can not only compete on their favourite games but also communicate in real-time using VoIP [38]; and *3G Mobile*, running over the IP Multimedia Subsystem (IMS) [54], [294]. A typical example of a communication service enhanced by VoIP is *location-independent telephony*, in which a user can own "local" GSTN (E.164) numbers (normally associated with landlines) in numerous different cities of the world, and receive phone calls, over IP, through any of these numbers, anywhere else, at normal GSTN prices. For instance, owning a telephone number in a particular city, a user can be reached from any GSTN phone in that city at the price of a local call and answer from any other city of the planet covered by the same VoIP provider (e.g., Skype), without any additional cost or the caller ever knowing the actual location of the callee.

3.8.4 VoIP Variants

In parallel to conventional IP Telephony systems, which assume a wired (and usually Ethernet-based) data link and physical layer, a number of VoIP variations have been proposed by the industry and are already widely implemented in special case or next generation networks.

Most current and forthcoming wireless voice and data communication technologies offer support for IP Telephony services (particularly VoIP), either exclusively, or as an extra provision. Such technologies are: Wireless Personal Area Networks (WPANs) conforming to the IEEE 802.15.x specifications (based on Bluetooth) [81], Wireless LANs (WLANs) complying to the IEEE 802.11 (WiFi) standards [81], [122], Wireless MAN (and LAN interconnection) networks based on the IEEE 802.16 (WMAN/WiMAX) standard [81], [280], as well as third (and fourth) generation mobile networks [54] and applications built for the IP Multimedia Subsystem [294], including standards produced by the 3rd Generation Partnership Project (3GPP) that also ratifies the second generation (2G) Global System for Mobile Communications (GSM) [U54], and the 3rd Generation Partnership Project 2 (3GPP2) [U55]. All these specifications define wireless data link and physical layers, over which IP multimedia can be transported, and use SIP for signalling and RTP for media encapsulation, similarly to conventional (wired) IP Telephony networks.

"Skinny" is a proprietary VoIP technology for wired or wireless VoIP, originally developed by Selsius Corporation and later on acquired by Cisco [220], [U7], which has progressively added it to a number of its own products, in parallel to support for standard signalling protocols like H.323 or SIP. Skinny operation is based on a lightweight stimulus mechanism, the Skinny Client Control Protocol (SCCP), used over TCP/IP for signalling between a client device (e.g., an IP Phone) and a softswitch like the Cisco Call Manager, which can then act as a proxy for connecting to larger scale IP Telephony networks, or even to the GSTN; in both cases, media transportation is conducted over RTP/UDP/IP, as in conventional

VoIP applications. Cisco's market penetration makes Skinny a popular, in terms of installation base, technology, which successfully addresses the increasing complexity of the main IP Telephony signalling mechanisms (H.323, SIP, MGCP and H.248/MEGACO) for simpler usage scenarios, like the typical case of a terminal connecting to the backbone network.

PacketCable [267] is a technology framework introduced in 1997 by CableLabs [U5], the cable operators' research consortium, for standardising the delivery of IP-based multimedia services over cable television networks. It is built on top of the Data over Cable Service Interface Specification (DOCSIS) standard (also known as the Cable Modem project) [U6] and includes provisions for voice transportation. In addition to DOCSIS (which implements the data link and physical layers) and IP (used at the network layer), the framework uses MGCP for signalling, which it extends in the form of the Network-based Call Signalling (NCS) protocol for controlling analogue residential gateways and the Trunking Gateway Control Protocol (TGCP) for Media Gateway control, although as of version 2.0 MGCP has been replaced by SIP. For media encapsulation, PacketCable uses RTP, like conventional IP Telephony networks.

Using IP for directly carrying narrowband services and TDM traffic is being examined by a number of vendors [15]. TDM over IP (TDMoIP) [352], in particular, is a transport technology for relaying native telephony channels over IP networks using pseudowires [45], transparently to individual signalling and traffic protocols. It comes in two variants, Circuit Emulation (CE) and Compressed Voice (CV). The CE service is implemented over MPLS [310], uses AAL1 [190] for traffic adaptation and maintains clock synchronisation in order to provide constant bit rate (T1/E1 or T3/E3), G.711 [181] voice channels over IP. The CV service carries variable bit rate voice with a choice of CODECs as well as fax signals over IP, uses AAL2 [191] for traffic adaptation and is very similar to conventional VoIP, but with transparent signalling and more efficient bandwidth utilisation. TDMoIP is an interesting idea for IP-based voice transportation and presents certain advantages (including simplicity of implementation and investment protection) for interworking with the GSTN, however it is a rather proprietary technique (mainly promoted by its inventor, RAD Data Communications [U40]) and is thus not widely deployed or field-proven.

Another variant of VoIP that will be increasingly of interest in the future relates to IPv6 [90], the next version of the Internet Protocol, which introduces significant improvements over the current version, IPv4 [296], including simplified header processing, an 128-bit address space, support for flow-based traffic, improved extensibility and enhanced security mechanisms, at the expense of doubling the minimum overhead (header size) from 20 to 40 bytes per packet, thus increasing the inefficiency of VoIP bandwidth-wise. IPv6 is still not widely deployed, but its standardisation process is fairly mature and, by design, the protocol is expected to fully replace IPv4 in the medium to long term [321]. This, in turn, means that current VoIP protocols (including H.323, SIP and RTP) will have to be updated or modified in order to work over IPv6 networks [269]. During the transitional period, special translation functions will be needed for interoperability between IPv4 and IPv6 Telephony networks [68].

Other advances in IP networking technologies may result in modifications to VoIP, as well. The Internet2 consortium, for instance, a group of over 200 universities managing "Abilene", a very high speed (currently 10 Gbps) IP backbone in the United States, operates a special VoIP Working Group for examining the implications of convergence technologies on voice communications. Similarly, IP-optical integration, through technologies like WDM and DWDM [124], will probably have an impact on VoIP standards, too. Finally, the addition of video (i.e., the transition to VVoIP), will have an impact on usage models, client application design and network resource availability, at least in the short to medium term, calling for a considerable improvement of existing tools and techniques [67], [70].

3.8.5 The Economics of VoIP

The economics of VoIP are mainly focused on two areas: pricing of calls and implementation costs. These, in turn, significantly affect the adoption rates of the technology by the user base and, hence, revenue generation for IP Telephony providers.

The issue of pricing in telecommunications networks is a complex one on its own, and becomes further complicated in the case of VoIP, due to its hybrid data/voice nature, the burstiness of the traffic carried and the lack of regulation. In essence, the communication service is treated as a commodity that is costly to produce but cheap to reproduce; in such a model, *units of service* can be defined, each of which is associated with an amount of money (its *price*), while a particular structure of prices constitute a *tariff*. A common way of calculating the charges associated with a tariff is the general form $\mathbf{V = a + bT + cL}$ [82], where a represents startup costs (e.g., a monthly fee), T is the duration of the communication, L is the distance of the endpoints interconnected by the network and b,c reflect the weight of each category in the total value, V (for example, it could be b = c = 0 for a flat-rate service, or c = 0 for a nationwide time-based service with a monthly line rental fee).

Price differentiation is generally based on the cost created and the level of service requested by the usage of network resources, while charging is based on the following parameters (or their combinations) [50], [82]:

- *Time.* The charge is proportional to the duration of communication.
- *Distance.* The charge varies according to how far away the communicating endpoints are from each other.
- *Volume.* The charge is calculated according to the amount of traffic load created by the calling user.
- *Congestion.* The charge reflects the statistical fluctuation of congestion in the network.
- *QoS.* The charge is determined according to the service level the user has requested from the network.
- *Subscription.* The charge is only a flat fee, which allows otherwise unlimited (or very high) usage of resources.

Like in the GSTN, pricing in IP Telephony networks can be exploited as a means of flow control, by providing (or removing) incentives for the use of resources in certain ways (e.g., up to a maximum volume, or at a particular time-of-day), thus allowing operators to roughly estimate or shape traffic profiles and engineer the network accordingly. The ETSI Open Settlement Protocol (OSP) [106], part of the TIPHON framework [107], can be used for the exchange of pricing, authentication, authorisation and usage information across different domains (e.g., ITADs or GSTN provider clouds), over XML [381], HTTP [110], or TLS [93]; OpenOSP [U36], an open source implementation of the protocol, is maintained by Vovida [U66].

While there is no standardised or universally adopted model for VoIP pricing, the subscription-based (flat rate) method is very popular [70], [294], [377], particularly for residential customers, where Internet access is combined with voice services at the same, fixed cost; this has obviously to be related, however, to time and distance-based charging, whenever calls to the GSTN are placed.

VoIP call charges are, in general, very attractive [70], [128], [377], but they do not always reflect the overall cost of implementing a VoIP network, particularly for home or business use. While technologies like softswitch allow carriers to substantially reduce their expenditure in deploying voice services by resorting to IP, as already discussed, purchasing and operational costs of new equipment necessary for VoIP may not always be justifiable for lower-end users. Most VoIP terminals, for instance, cannot be powered from the network, as normal telephones, resulting in much higher electricity bills which, for large organisations, may become prohibitively high, compared to per-minute cost savings [72], [295]. The aggressive pricing adopted by traditional telcos due to market deregulation and competition, exacerbates this fact.

3.8.6 VoIP Adoption

Voice over IP is competing for market share against a technology (conventional telephony) that has been well-established since the late 19th century [26], both at a user acceptance and at a technical level, therefore its adoption rates will depend on its comparative performance in both of these areas. Billions of people worldwide are already accustomed to the simple and reliable user interface of the GSTN (i.e., the telephone keypad), and, as Information Technology adoption research shows, attempts to convert them to more complex, computerised devices (like PC-based softphones, or advanced IP Phones), can result in a restriction of the VoIP subscriber base to the small minority of highly technically skilled or innovative people (unless, of course, VoIP becomes a disruptive technology, which does not seem to be the case, despite the initial overhype) [70], [159], [276], [309]. In other words, cost and efficiency are not, on their own, strong enough factors for switching from the GSTN to IP Telephony [295], [377]; rational expectations in terms of performance (e.g., acceptable QoS, as already discussed) and economics (longer-term return on investment) must also be cultivated among the public for VoIP to globally succeed [14].

3.9 Interoperability with the GSTN

The GSTN has several "philosophical" differences compared to IP Telephony, beyond the circuit and packet switching dichotomy [253]. As seen already, the two technologies differ substantially on several aspects, including network intelligence (centrally located for the GSTN vs. distributed for VoIP), performance (QoS), reliability (equipment and link lifetime), availability (uptime), scalability (including call support and upgradeability), efficiency (bandwidth utilisation), functional specification (state machines), addressing (E.164 numbers vs. text-based aliases), timing (synchronous vs. asynchronous), traffic profiles (constant vs. variable bit rate), regulatory status (including emergency services), and pricing policies (particularly for long distance communication) [72]. From an architectural standpoint, the GSTN has a hierarchical structure with most functionality integrated in a small, tightly related set of standards; by contrast, IP Telephony functionality is implemented in a modular way, by a combination of protocols, which inevitably has an impact on performance [74]. All these differences reflect on the effectiveness of GSTN-VoIP interworking solutions and complicate their implementation.

Transparent end-to-end connectivity across GSTN and IP Telephony networks is achievable, as already discussed, via installing special "translation" functions (gateways) at interconnection points. In addition to standard signalling and media translation, replication of long-established Intelligent Network (IN) [120], [203] and Next Generation Networking (NGN) [209], [228] services, as well as support for already standardised supplementary ones (like those described in H.450 [189]) is desirable, however typical gateway implementations tend to focus on core functionality and offer extra features depending on the target user base. Depending on the architecture of the network and the topology of the call scenario, more than one gateway may be needed end-to-end, for joining together different network "islands" and even for translating between incompatible GSTN and VoIP protocols (e.g., H.323 vs. SIP [160]).

VoIP-GSTN interworking scenarios have been studied by all major telecommunications standards bodies. The ETSI TIPHON group [107] (part of the TISPAN Technical Committee as of September 2003) has produced a comprehensive framework that addresses all major aspects of interoperability, from simple connectivity to QoS and security mechanisms, and is independent of signalling protocols, i.e. it can be implemented using H.323, SIP or H.248/MEGACO. The ITU-T covers several different interworking scenarios between circuit and packet switched networks under its E, H, X and Y series of Recommendations [208], with particular emphasis on GSTN-H.323 interoperability [188]. Similarly, the IETF is focusing on GSTN-SIP interworking [378], [401] and, in addition to individual standards, has developed a number of frameworks for this purpose: PINT [292] covers the invocation of certain telephone services (e.g., call back) from the Internet; SPIRITS [140], [347] supports services (e.g., call waiting or caller ID) originating in the GSTN and requiring interaction with a VoIP network; finally, SIGTRAN [282] addresses issues related to telephony signalling transportation, performance and interoperability over IP, within GSTN requirements.

3.10 Large-Scale VoIP

The scalability (in terms of number of installations and amount of calls carried worldwide) of IP Telephony will depend, as already discussed, on both social and technological factors. On the first front, widespread adoption by end users will be required; for the second area, the performance of both the software (protocols, applications) and the hardware equipment is important, with more emphasis necessarily going to the design/architectural part, given the high amounts of low cost processing power already available.

Social factors (including ease of use, pricing, and other technology adoption parameters) can be resolved by following the well-proven GSTN usage model [26], [49], [145], [146], [307], [325] either natively (i.e., by making VoIP terminals and the whole service look and operate like conventional telephony), or in the backbone (i.e., using the GSTN for access and carrying voice over an IP infrastructure at the core of the network). Security, QoS and interoperability with the existing telephone network will have to be offered at acceptable service levels to subscribers before universal VoIP can be realised and, while technologies for achieving this objective already exist as seen above, their application in large-scale (geographically and in user numbers) IP Telephony networks remains to be seen. For the same reasons, successful integration of both legacy networks and equipment will be required [315], and this can add further challenges to creating a global VoIP infrastructure.

On the technical front, the issue of scalability surfaces at both the control plane and the user plane. *Signalling* is a connection-oriented process, in the sense that state must be maintained (e.g., for call accounting and billing), as well as because of the reliability required in the exchange of control messages. All major IP Telephony signalling protocols, including H.323, SIP and H.248/MEGACO, can run over TCP for this purpose, and need to add various amounts of state in their corresponding servers, thus reducing their scalability, compared to connectionless/stateless architectures [357]. Pushing the intelligence towards the edges of the network according to the end-to-end argument [324], as happens with SIP [53], can help alleviate the situation but this is doable to a limited extent, due to the VoIP operational model, as already explained. *Interactive media transport* is less challenging, as it happens over RTP and (connectionless) UDP for point-to-point calls, but more advanced services, such as conferencing, may present scalability problems, too. Moreover, there are differences in efficiency among the various protocols in the first place: SIP, for instance, being text-based, tends to produce large messages and, thus, consume more bandwidth than H.323, thus scaling to a smaller amount of simultaneous calls if all other communication parameters are kept the same.

Call routing is another potential bottleneck in the deployment of large-scale IP Telephony networks. In such architectures, for numerous reasons (like functionality, security and commercial policies), certain devices must be present in the end-to-end signalling and/or media path of each call, thus increasing processing delays and creating single points of congestion or failure. Such devices include gatekeepers in the routed H.323 model [159], SIP Proxy Servers [216], Session Border Controllers [158], firewalls and NAT routers [128], as well as other "middleboxes" [351]. *Gateways* are also a major source of scalability problems, since they need to allocate significant computing power in order to translate between GSTN and VoIP formats, work in a mostly connection-oriented/stateful mode and serve large amounts of users transparently.

Decomposed *gateways*, following the softswitch model, are, as seen above, a proposed solution to VoIP scalability problems. On the control plane, softswitch implementations can already support up to tens of thousands of calls [281], on a par with GSTN Class 4 and Class 5 switches, however they must rely on banks of Media Gateways on the user plane, as current MGs can only offer hundreds or thousands of ports each [159], i.e. one order of magnitude less than softswitches. User Agents (softphones or IP Phones), on the other hand, need to support no more than a few simultaneous calls (for conference connections), if not just one. These figures, summarised in Table 3.1, suggest that, to cover large urban centres, populated by $O(10^6)$ people and thus in need of supporting a maximum of $O(10^7)$ different calls (to cater for the various types of telephony), either $O(10^3)$ (i.e., thousands) of softswitches or signalling servers must be provisioned, or a more hierarchical architecture should be adopted. The latter approach is successfully used in the GSTN and has also been proposed for data networks already since the 1970s [225], [226].

VoIP Device	**Scalability (supported calls)**
Softswitch or Signalling Server	$O(10^4)$ - $O(10^5)$
Media Gateway	$O(10^3)$
Softphone	$O(10^1)$

Table 3.1: Approximate scalability of IP Telephony devices

In addition to architectural decisions like the softswitch-based operation, the scalability of VoIP networks can also be enhanced by traffic engineering solutions, based either on voice (telephony), or data (IP) techniques. For the first case, it can be reasonably assumed that commercial VoIP networks will implement some type of call admission control and capacity limitation, thus they can be approximated by circuit-switching models, allowing teletraffic calculations (e.g., Erlang and Engset formulas) to be applied in their dimensioning [160], [175]. IP-specific traffic engineering solutions, on the other hand, are based on QoS (particularly DiffServ and MPLS) [111], [224], or common routing protocols [115], [301]. Traffic monitoring can also be used for enhancing IP Telephony network scalability by dynamically adjusting traffic paths and the IETF has extended RTP to provide the additional information needed for this purpose [118].

3.11 Conclusion

As IP Telephony continues to expand towards the ultimate goal of offering the same level of features with the GSTN and new, advanced services in addition, it faces a growing number of technical challenges in areas such as user adoption, QoS, reliability, security, and, crucially, interoperability with existing telephony networks. Successful solutions to these issues will determine, to a large extent, the applicability of a universal IP Switched Telephone Network (the ISTN), as a successor to the current mix of GSTN and IPTNs in operation worldwide.

Complementing the general critique of packet voice presented in Chapter 2, this chapter has overviewed the current state of the art of technologies developed specifically for IP Telephony, with additional emphasis on the solutions necessary in the design and implementation work that follows. The preceding discussion reveals that the VoIP scalability problem investigated in this thesis is a multifaceted one, with social and technical dimensions, both of which are addressed in the investigation of the core components (the user agent, the gateway and the call routing mechanism) described and evaluated in Chapters 4-7.

CHAPTER 4

A Framework for

Large-Scale IP Telephony

4.1 Overview

The Internet Protocol, its cooperating standards and the networking philosophy associated with them, are usually seen as the exclusive path towards IP Telephony. Despite the attractiveness of an all-IP voice network, however, the GSTN model should not be dismissed altogether. On the contrary, there are lessons to be drawn from its exemplary performance in terms of universal connectivity, reliability, scalability, QoS and features - all of which remain targets for VoIP networks. Much of these achievements can be credited to the equipment and routing stability of the GSTN, which are largely incompatible with the best-effort philosophy of IP and therefore tend to be absent from the design of VoIP standards.

In this chapter, a new *Large-scale IP Telephony Framework* (LIFT) is proposed and evaluated. The framework is based on the "call anyone, anywhere, anytime" model of the GSTN, but maintains an IP bias and aims at supporting global connectivity by encompassing existing circuit-switched and packet-switched networks, while adopting benefits from both of these networking worlds. Core to LIFT is the generalised concept of a gateway, the definition of which is influenced by the GSTN operational model. Characteristic parts of the communications architecture described by the framework are implemented and analysed in Chapters 5, 6 and 7.

Framework	Description
TIPHON	Thoroughly specified; defines the concept of a decomposed gateway
ITU	Based on H.323; extensive set of accompanying standards
IETF	Based on TRIP and SIP; loosely specified
Softswitch	Preferred architecture for large-scale IP Telephony
Skype	Based on P2P networks; over 100 million users worldwide
LIFT	Defined in this thesis; a new approach to large-scale IP Telephony

Table 4.1: IP Telephony frameworks

4.2 Related Work

A number of existing technical standards, such as the ETSI TIPHON [107], the ITU-T (based on H.323 [188]), the IETF (based on TRIP [314]), the emerging softswitch model [281], and, to an extent, the Skype P2P network model [21] define, implicitly or explicitly, the architecture of a large-scale IP Telephony network, to varying degrees of completeness (Table 4.1).

The TIPHON group has probably generated the richest set of specifications thus far, covering virtually all aspects of communication, from interworking scenarios to QoS, security and pricing [107]. TIPHON has introduced the model of the decomposed gateway that has been adopted by all other frameworks (including LIFT, albeit in a modified way, as explained later on) and is considered a complete specification by the ETSI; hence, the project has been migrated to TISPAN (Telecommunications and Internet converged Services and Protocols for Advanced Networking) after version 4, instead of proceeding independently to the originally planned version 5. TISPAN is defining the full architecture of a Next Generation Network, of which Release 1 [108] is based on similar work from 3GPP [U54] and 3GPP2 [U55].

The ITU-T IP Telephony framework is centred on the H.323 protocol [188] and accompanying standards of the H.3xx series of specifications that collectively represent a very mature architecture, covering a similarly broad area of communication requirements and scenarios as TIPHON/TISPAN. The H.323 network model tries to combine the benefits of both circuit switching and packet switching and, although it is purely packet-based (supporting not only IP, but also ATM and other protocols), the influence of the GSTN legacy in the ITU has added not only to its complexity, but also to its versatility and scalability [159]. Medium to large size IPTNs running H.323 are gatekeeper-controlled, which, as discussed in Chapter 3, makes them approximate the behaviour of capacity-limited SCNs and thus assisting their performance in areas like overall Quality of Service [156]. H.323 is currently in version 5, and a version 6 is already in the works within the ITU-T [U21].

IETF's approach to defining the architecture of an IPTN is also focused on its main signalling protocol, SIP [318], expanded by additional standards like MEGACO [134], but the specification completing the puzzle is the TRIP framework [316]. Due to the IETF philosophy of having (at least) one protocol for every different task, these standards are not fully integrated and also draw heavily from the TIPHON and H.323 network models, particularly when it comes to interworking with the GSTN. For such cases, as seen in Chapter 3, TRIP only covers the scenario in which calls are originated from an IPTN and terminated in the GSTN, and makes further generalisations, on the (rather simplified) assumption that the other possible communication scenarios can be reduced to trivial IP connectivity cases.

The softswitch model [281], [391] is not, strictly speaking, a fully specified framework on its own, however it merits separate consideration because of its universal adoption as the architecture on which large-scale IP Telephony networks can be built. All other frameworks (including LIFT) accept the concept of employing highly powerful devices that implement the functionality of complex and expensive Class 4 and Class 5 GSTN switches in software and then using such devices (i.e., softswitches) for controlling banks of (not necessarily collocated) Media Gateways, over IP. In a sense, the softswitch model can be seen as an evolution of the H.323 gatekeeper model, which also draws heavily from circuit switching. This fact, combined with the universal adoption of softswitches, adds an implicit GSTN bias to all IP Telephony frameworks, thus justifying the part of the hypothesis of this thesis positing that the advantages of the GSTN should not be overseen when designing current or next generation IPTNs.

The Skype [U48] network model represents another architecture which can be considered as a (partial) IP Telephony framework, since it includes all the basic components found in other frameworks and only differentiates itself because of the proprietary mechanisms it employs for signalling, media and call routing. More specifically, Skype uses a distributed P2P grid based on KaZaA [U25], where terminals (called *ordinary nodes*) are personal computers or PDAs running a version of the Skype application, signalling servers (called *supernodes*) are hosted in end user PCs with enough available processing power and bandwidth, gateways exist for supporting calls to or from the GSTN (for the SkypeOut and SkypeIn services, respectively), and the necessary authentication servers are centrally managed by the company itself [21]. In that respect, the Skype model has similarities to the LIFT-compliant HIT mechanism for call routing, discussed in Chapter 7. Skype's distributed architecture works surprisingly well in terms of both reliability and overall QoS, resulting in the creation of a worldwide user base that currently exceeds 100 million, out of which around 5% are simultaneously online, as already seen in Chapter 3; thus, the Skype network actually models, in real-time, the voice communication conditions that can be found in a country with a population comparable to that of Britain, but distributed over a much wider geographic area. Beyond a high quality voice service using wideband CODECs, the application supports instant messaging and file sharing, while also introducing video services, hence shifting its attention towards VVoIP [U48].

A detailed comparison of the core LIFT component, the gateway, and of the entire framework, to other similar specifications, is presented later in the chapter, where similarities to the SS7-based GSTN operational model are also revealed. More specific details about the technologies discussed in this section and the rest of the chapter can be found in the corresponding parts of the literature review, presented in Chapters 2 and 3.

4.3 Definitions

A number of terms are defined in the context of a LIFT-compliant IP Telephony network. Those common with other frameworks are usually perceived the same way for LIFT, too, but there are occasional differences which, in combination with some new concepts, mandate the following, detailed alphabetical listing.

- **Address**: An IP Telephony address (or simply address) is a GSTN (e.g., E.164) address, an IP (IPv4 or IPv6) address, or any other routable (i.e., structured and globally unique) identifier (e.g., a domain name), specified and usable in the context of an IP Telephony network.
- **Attribute**: A routing parameter considered individually (on a per-hop basis) during path calculation. For instance, the formats supported by a Media Gateway could be considered values of a relative attribute.
- **Call packet**: An IP-encapsulated Protocol Data Unit (PDU) containing signalling or media information relative to an IP Telephony call in progress.
- **Call (routing) path**: An application-to-application path of IP Telephony gateways that interconnect the source and the destination terminals of a call, along with those terminals. This path is defined at the application layer, as network layer routing is performed exclusively by the IP protocol.
- **Call route**: The set of all paths joining two IP Telephony terminals.
- **Call Router (CR)**: A device with IP connectivity that stores, processes and advertises call routes, but not call packets.
- **Constraint**: A factor restricting the results of a call routing path calculation. Constraints can be either attributes or metrics. The term is used in an identical manner with the term *routing parameter*, defined later on in this section.
- **Gateway location/gateway selection/call routing**: The process of identifying a gateway during the formation of a call path from a source to a destination terminal [315]. This procedure usually involves much more than simple address resolution, so the term "gateway selection" is perhaps more suitable than "gateway location". Nevertheless, both expressions are used interchangeably in this document to refer to the application layer part of the overall IP Telephony call routing problem.

- **Host**: Any IP Telephony call router, gateway, terminal, or other application layer device (e.g., a signalling server) participating in an IP Telephony session.
- **IP Telephony Administrative Domain (ITAD)**: A set of IP Telephony hosts or networks of hosts that are subject to the same administrative policy.
- **IP Switched Telephone Network (ISTN)**: The universal collection of directly or indirectly interconnected IP Telephony hosts, within which any host can eventually communicate with any other, subject to administrative policies. The ISTN is a global scale IPTN integrating circuit and packet switched networks.
- **IP Telephony Network (IPTN)**: Any network capable of offering IP Telephony services to its hosts.
- **IP Telephony User Agent (IPTUA)**: A software application, which allows users to place and receive calls via an IPTN. An IPTUA can be implemented in a number of different ways, including an *audio tool* and a *softphone*.
- **Link**: A wired or wireless connection joining two IP Telephony hosts.
- **Metric**: A *routing parameter* considered cumulatively (on a per-path basis) during call path calculation. For example, the end-to-end delay of a call could be viewed as the value of such a parameter.
- **Routable Address**: An identifier which is both globally unique and structured, so that it can be allocated according to actual network topologies.
- **Routing parameter**: A factor restricting the results of a call routing path calculation. Routing parameters can be either attributes or metrics. The term is used in an identical manner with the term "constraint", defined above.
- **Signalling Server**: A device that performs call control (i.e., signalling) functions, including setup and clearing of calls, admission control, address translation, location registration for mobility, maintenance of Call Data Records (CDRs), as well as Authentication, Authorisation and Accounting (AAA), wherever applicable. The Signalling Server is the application layer device responsible for routing call control packets from source to destination.
- **Softphone**: A software application running in a computer or similar device, with appropriate signalling and media capabilities so that it can place and receive IP Telephony calls. A softphone is a type of IPTUA.

- **Softswitch**: A software application running in powerful (general-purpose or dedicated) computer and communications hardware that replicates the signalling functionality of advanced telephony switches.
- **Terminal**: A GSTN or IP device, such as a telephone or a workstation, offering IP Telephony services to one or more end users.

The above terms are further clarified as they are used in the rest of this chapter.

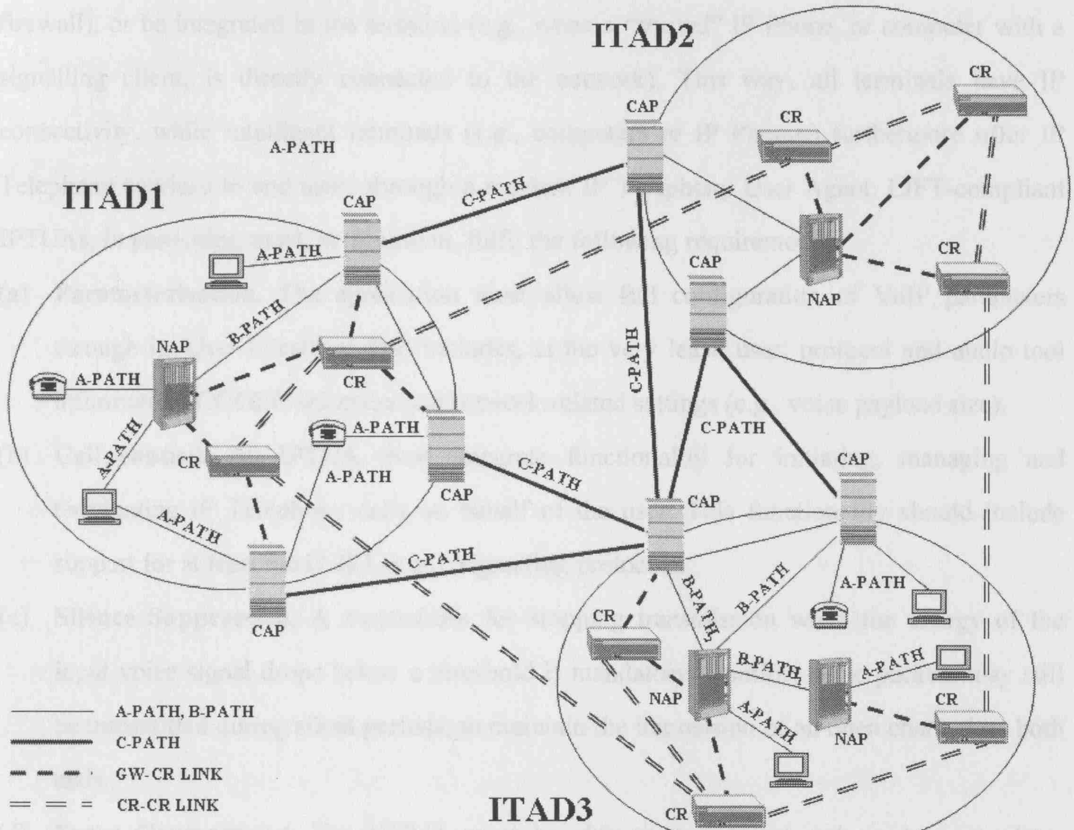


Figure 4.1: Application layer view of a LIFT-compliant IP Telephony Network

4.4 Network Architecture

A simplified, Application Layer view of a LIFT-compliant IP Telephony Network is presented in Figure 4.1, where the IPTN described is overlaid on top of a conventional IP network, which acts as the carrier mechanism. The details of this architecture are presented in detail in the following sections.

4.4.1 Terminals

Every IP Telephony Network consists of a (potentially large) number of end-user access nodes, its terminals, which can be anything from legacy telephones and new IP Phones, to intelligent wireless devices (e.g., PDAs) and sophisticated multimedia computers.

Terminals are physically connected to the IPTN via a gateway function, which can reside on a separate device (for example, in cases of a telephone or a workstation that is behind a firewall), or be integrated in the terminal (e.g., when a "trusted" IP Phone, or computer with a signalling client, is directly connected to the network). This way, all terminals have IP connectivity, while intelligent terminals (e.g., computers or IP Phones) furthermore offer IP Telephony services to end users through a resident IP Telephony User Agent. LIFT-compliant IPTUAs, in particular, must, at minimum, fulfil the following requirements:

(a) **Parameterisation.** The application must allow full configuration of VoIP parameters through its User Interface. This includes, at the very least, user, protocol and audio tool information, CODEC selection and network-related settings (e.g., voice payload size).

(b) **Call control.** An IPTUA must integrate functionality for initiating, managing and terminating IP Telephony calls on behalf of the user. This functionality should include support for at least the H.323 or SIP signalling protocols.

(c) **Silence Suppression.** A mechanism for stopping transmission when the energy of the input voice signal drops below a threshold is mandatory. Comfort noise packets may still be transmitted during silent periods, to maintain the impression of an open channel on both ends.

(d) **Error Concealment.** The IPTUA must be able to understand and react to any loss, sequencing and duplication problems affecting incoming voice packets, so that the effect of those problems is hidden from the user, whenever possible.

(e) **Media processing.** The capturing of voice through the audio interface, for transmission purposes, as well as playback of received voice packets while maintaining timing dependencies. Full RTP support is compulsory for the implementation of this requirement.

4.4.2 Signalling Servers

IP Telephony networks, as discussed in Chapter 3, operate numerous logical or physical entities handling control operations. Signalling servers, in particular, perform most of the call routing task (i.e., the routing of the signalling call packets) and are thus indispensable, but in medium to large size installations they are not usually stand-alone; instead, they tend to be collocated with other similar functions, in the same hardware. Thus, a commercial product implementing signalling is closer to a full scale softswitch, than to an individual H.323 gatekeeper, or SIP Proxy Server [281]. Softswitches, in turn, are often associated with one or more Media Gateways in their networking vicinity, thus collectively forming full-function gateways, or analogous, larger scale distributed devices [79].

Based on the above, LIFT perceives signalling servers and other similar control entities as varieties of softswitches and those, in turn, as special cases of gateways, as analysed in the following section.

4.4.3 Gateways

IP Telephony gateways are application layer devices that increasingly acquire more functionality, in order to accommodate a wide variety of interoperability and performance scenarios [79], much like network layer gateways (i.e., routers), which often integrate multiprotocol, wireless, bridging and switching support [289]. Accordingly, LIFT assumes that VoIP gateways can mutate from simple signalling or media components limited in local network environments, to highly scalable distributed devices supporting thousands of calls from anywhere in the planet. This represents a "generalised" view of the gateway concept.

In accordance with the above, LIFT gateways must, at minimum, comply with a number of functional requirements:

(a) **"Black box" operation.** Every gateway should operate as a virtual "black box", encapsulating signalling and media components, to transparently connect two (or more) potentially different voice carrier sub-networks. Whenever an external event (e.g., a reset request, or congestion) triggers internal reactions (e.g., switching to lower-rate media encodings and eventually re-routing of packets from one component of the gateway to the other), the effects of such actions should be confined as much as possible inside the gateway itself.

(b) **SCN connectivity.** Each gateway must have full connectivity, through at least one physical connection and address, to an SCN network (typically, to the GSTN).

(c) **IP connectivity.** Each gateway must have full connectivity, through at least one physical interface and IP address, to at least one IPTN.

(d) **SCN-IP interworking.** Gateways must be able to offer full and bidirectional SCN-IP interoperability. This means that for every call, irrespective of where it originated, the gateway must accept as its input call packets from the source, process them as required, produce a series of corresponding call packets as its output towards the destination, and vice versa.

(e) **Support for SCN-only or IP-only connections.** Gateways must be able to support calls where the interconnected sub-networks involved are operating exclusively on SCN-only or IP-only protocols. The existence of such gateways may be necessary depending on application and network configurations. Some typical examples are "patch-through" connections of two SCN channels through a gateway, media conversion, transcoding, mixing or relaying packets of a conference session, or a point-to-point call between two audio tools.

(f) **Multipoint Call Facility.** The gateway must provide for all types of call connection scenarios, point-to-point, point-to-multipoint and multipoint-to-multipoint, both at the signalling and at the traffic (media) level.

(g) **Distributed implementation.** Every gateway must be constructed as a collection of internal components that operate as a loosely controlled network. The actual number of these components can be arbitrarily large, pending performance and scalability considerations. Communication among the components of the gateway must happen over IP, via an appropriate application layer control protocol, or a similar mechanism.

(h) **Terminal access.** Suitable gateways must act as points of access to the network for terminals, and every terminal must access its IPTN via a gateway. This allows both SCN and packet terminals to be reachable via IP.

(i) **Chaining.** In the real world, it is not uncommon for call packets to cross any number of signalling entities, such as Signalling Servers (e.g., a chain of SIP Proxies, or a device converting from Q.931 to H.323 to SIP, when there is no direct connection/conversion). The same conclusion applies for the media path, where, for example, conversion from A-law PCM to MPEG-1/2 Audio Layer 3 (MP3) and then to u-law PCM may be needed for a particular call. Of course, "transcoding" situations like this should be avoided as much as possible, because of obvious real-time media constraints; however, when present, they must be handled transparently to the user, so gateways must provide signalling and media capabilities that allow tandem connections.

Implementing the above requirements, the abstract architecture of a "generalised", LIFT-compliant IP Telephony gateway is materialised as shown in Figure 4.2 and analysed in the following sections.

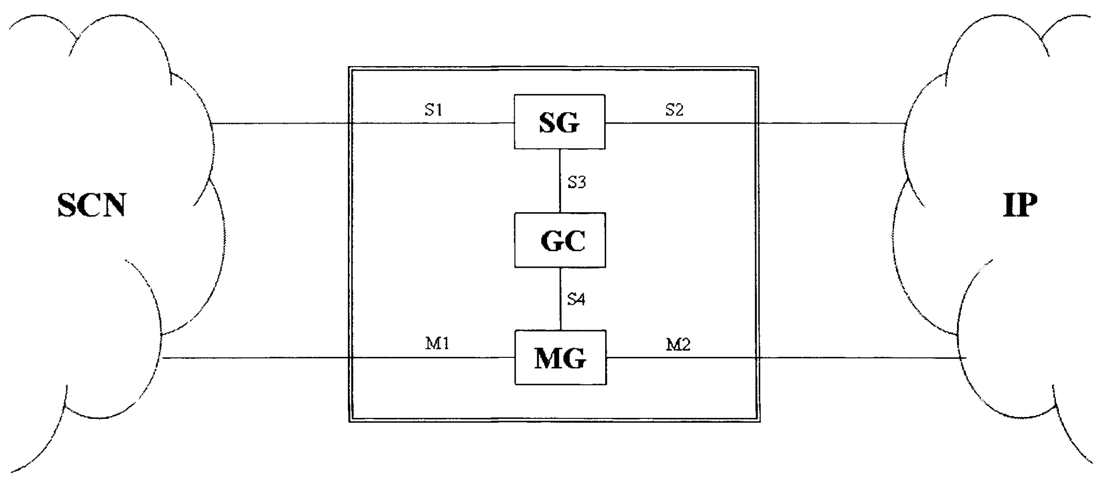


Figure 4.2: Architecture of a LIFT gateway

4.4.3.1 Gateway Components

The basic functional components of a LIFT gateway are its Signalling Gateway (SG), Media Gateway (MG) and Gateway Controller (GC). In addition, there are a number of interfaces, interconnecting the components inside the gateway with the external networks surrounding it.

Every gateway is built around a single GC, which constitutes its core (master) component; depending on desired functionality, the gateway may also include zero or more SGs and zero or more MGs, all of which operate in a master/slave fashion under the instructions of the GC. If the gateway needs to have more than one GCs (usually for redundancy reasons), one will be acting as the primary, in order to maintain the master/slave operational model. If the gateway has one or more SGs but no MG, it will offer only signalling services to the IPTN where it belongs. If it has no SGs but one or more MGs, it will offer only media services. In any case, the exact number of GCs, SGs and MGs will be defined by the desired functionality, performance and reliability levels, since there may need to be multiple instances of a component, for scalability, redundancy and load balancing purposes.

4.4.3.2 Signalling Gateway (SG)

A Signalling Gateway is the component responsible for signalling adaptation among the networks connected by the gateway to which it belongs.

The SG communicates with the SCN network via its external signalling interface (S1 in Figure 4.2), which will usually be a GSTN link (e.g., an SS7 channel or trunk), and with the IP network via its external signalling interface S2. The SG also communicates with the GC via an internal, IP-based signalling interface (S3 in Figure 4.2), which allows the GC to control the SG in a master/slave fashion.

Typical operational scenarios for an SG are:

(a) *Heterogeneous Interworking* (translation), performed between SCN and IP subnets, such as in the case of a Q.931 to H.323 call.

(b) *Homogeneous Interworking*, connecting native SCN (e.g., for a GSTN-to-GSTN call) or native IP subnets (e.g., for an H.323 to SIP call).

(c) *Signalling Adaptation*, in which the SG acts as an edge or an intermediate Signalling Server (e.g., a SIP Proxy Server). From this point of view, any IP Telephony Signalling Server can be viewed as a (simplified) SG.

Signalling Gateways are "weak" network components, meaning that they participate in the operation of an IPTN only in association with and under the control of the gateway in which they belong.

4.4.3.3 Media Gateway (MG)

A Media Gateway is the component responsible for media adaptation among the networks connected by the gateway to which it belongs. Its operation is similar to that of a Signalling Gateway, but for media instead of signalling.

The MG communicates with the SCN network via its external media interface (M1 in Figure 4.2), which will usually be a GSTN link (e.g., an ISDN BRI, or a T1/E1 trunk), and with the IP network via its external media interface M2. The MG also communicates with the GC via an internal, IP-based signalling interface (S4 in Figure 4.2), which allows the GC to control the MG in a master/slave fashion.

Typical operational scenarios for an MG are:

(a) *Heterogeneous Interworking* (translation), performed between SCN and IP subnets, such as in the case of an ISDN BRI channel, carrying TDM voice, to an RTP/UDP socket, carrying packetised voice.

(b) *Homogeneous Interworking*, connecting native SCN (e.g., for a GSTN-to-GSTN call) or native IP subnets (e.g., for an RTP-to-RTP call).

(c) *Media Adaptation*, in which the MG acts as an edge or an intermediate Media Server (e.g., an H.323 Multipoint Processor). From this point of view, any IP Telephony media application (such as a CODEC or a buffering/synchronisation function) can be viewed as a (simplified) MG.

Media Gateways are "weak" network components, meaning that they participate in the operation of an IPTN only in association with and under the control of the gateway in which they belong.

4.4.3.4 Gateway Controller

The Gateway Controller is the device that implements call control functionality inside a gateway, like similar entities (call agent, call broker, call manager) operating in monolithic GSTN switches. The main functions of a GC are:

(a) **Advertising the capabilities of the gateway to the IPTN**. This is done by a mechanism associated with the call routing protocol (e.g., by running a TGREP client [17] in a TRIP network [314], as discussed in Chapter 3). Internal capability collection by the GC can be implemented via protocols such as SLP [141], LDAP [384] and SIP (for example, using modified REGISTER requests) [318], or any other suitable method.

(b) **Monitoring the operation of the gateway components.** This is achieved by bidirectional communication between the GC and the SGs and MGs of the gateway (for example, using periodic H.323 RAS RRQ messages with the KeepAlive parameter set [188]).

(c) **Controlling the operation of the gateway components.** For every incoming call, the SG contacts the GC for a permission to forward the signalling packets (after translation, if needed). If the permission is granted, the SG will contact the MG after receiving notification (e.g., a Q.931 CONNECT message) that the called terminal has accepted the call. The GC will then instruct the MG to open the corresponding media channels and will continue to monitor the progress of the call until it is completed and according to the implemented call intelligence. The communication of the GC with the SGs and the MGs can be achieved through a variety of mechanisms, either custom, like a Proprietary Device Control (PDC) function, or more standardised ones, such as MBUS [284], MGCP [5] and H.248/MEGACO [136], [187] (the latter two specified by the IETF specifically for the GC-MG communication).

(d) **Implementing call intelligence in the gateway.** The GC acts as the decision centre of the gateway, so it is responsible for administration (including admission control, call control and billing) and other management operations, probably using proprietary (vendor-specific) implementation mechanisms.

Despite the breadth of potential roles described above, the GC is not necessarily a complex entity. For instance, it can be a very simple function that informs call routers within the IPTN of the existence of the gateway and the services it can offer, or just registers its internal components with a location service. As far as the network is concerned, GCs are only responsible for the presentation of their respective gateways as externally unique entities, and they do not participate in the call paths.

A gateway can participate in an IPTN even if it consists (perhaps temporarily) of only its Gateway Controller. This can happen, for instance, when the SG(s) and the MG(s) of the gateway are out of service for a period of time (e.g., re-booting). The GC then advertises the gateway's presence with a null set of signalling and media capabilities, although other features, such as location registration, may still be available. Furthermore, multiple GCs can operate inside a gateway for scalability, redundancy and load balancing purposes; in such an event, one of these will necessarily be internally elected as the primary (for example, following the H.323 master/slave election process) and will thus become the entity responsible for offering the described functionality to the gateway it belongs.

4.4.3.5 Gateway Interfaces

There are two types of interfaces in a gateway:

(a) External interfaces (such as the S1, M1 and S2, M2 pictured in Figure 4.2) connect the gateway to an IPTN and, depending on the operational scenario, can be either SCN or IP-based. The SCN-based ones will usually terminate GSTN signalling (e.g., Q.931 or SS7) or media (e.g., TDM voice), whereas their IP-based counterparts will be used to carry IP signalling (H.323 or SIP) or media (RTP-encapsulated) packets.

(b) Internal interfaces (such as the S3 and S4 in Figure 4.2) provide for connectivity among the gateway components and are exclusively IP-based. They are signalling channels, used for communication of the GC with the components of the gateway. The exact IP-encapsulated application-layer protocol carried over these links depends on the functionality implemented by the GC each time, as described above, however the most typical cases are the GC-SG communication (interface S3 in Figure 4.2), part of which is already specified in the IETF SIGTRAN family of protocols [282], and the GC-MG communication (interface S4 in Figure 4.2), which is standardising around the MGCP and H.248/MEGACO protocols, as already discussed.

4.4.3.6 Special Types of Gateways

Gateways are often classified according to the type of service they provide [316]. For the purposes of LIFT, in particular, two special types are defined, following the same categorisation principles.

4.4.3.6.1 Network Access Point (NAP)

A Network Access Point (NAP) is a gateway serving as the entry point of an end user terminal to an IPTN. For "dumb" terminals (such as POTS devices), a NAP is always required for signalling and media adaptation. For "intelligent" terminals, the associated NAP may as well be "null" (i.e., the terminal connects directly to the IPTN) or (more probably) a thin software layer (for instance, a lightweight H.323 client, or a Skinny client). However, there are cases, particularly in commercial environments, where a separate NAP will be needed/required even for this type of terminals (e.g., for admission control, or passage through a local IP firewall or Session Border Controller).

NAP operation is topology and call-related. Any gateway with proper technical characteristics can serve as a NAP for any terminal within an ITAD, and each terminal of an ITAD may be "plugged" to different NAPs within that ITAD, subject to administrative policies (ITAD1 in Figure 4.1 constitutes an example). Conceptually, a NAP implements the UNI part of an IPTN for the terminals it services, much like Access Gateways defined elsewhere [134], [136], [253].

4.4.3.6.2 Carrier Access Point (CAP)

A Carrier Access Point (CAP) is the gateway serving as the exit point of an end user terminal from an IPTN.

CAP operation is topology and call-related. Any gateway with proper technical characteristics can serve as a CAP for any terminal within an ITAD, and the terminals of an ITAD can place calls to destinations outside that ITAD using different CAPs of it, subject to administrative policies. Conceptually, a CAP implements the NNI part of an IPTN. Trunking Gateways defined elsewhere [134], [136], [253] could, among others, be used as CAPs.

4.4.3.7 LIFT Gateways compared to Conventional VoIP Gateways

Like LIFT, a typical VoIP gateway complying with the prevailing TIPHON [107] model used by all other IP Telephony frameworks, is built from three basic components, the SG, the MG and the MGC, as seen in Chapter 3. However, there are notable differences between the LIFT and TIPHON gateway architectures.

More specifically, a comparison between the two models reveals that:

(a) Both LIFT and TIPHON allow decomposed gateways, that is, devices in which the internal components do not have to be collocated; instead, they can be spaced apart and operate as a unified device via their interconnection over an IP network.

(b) The Signalling Gateway component in LIFT, in addition to TIPHON functionality, also performs the translation of SCN signalling to IP signalling. (This operation is the duty of the MGC in the TIPHON model, as already discussed.)

(c) The Media Gateway component is identical in both models and assigned with all media-related operations, under the control of another internal device of the gateway (the GC).

(d) The TIPHON Media Gateway Controller corresponds to the LIFT Gateway Controller component, in that they both implement call control functionality and manage the MG in a master/slave fashion. However, the GC delegates signalling translation to the SG, which it also controls in a master/slave fashion, much like the MG, allowing for a more symmetric operation; furthermore, the GC is assumed responsible for internal gateway capability discovery and advertisement to neighbouring call routers, OAM functions, billing, as well as all other call control operations. So, while for small scale implementations the difference between an MGC and a GC will mainly be the location of the SCN-IP signalling translation function, in larger networks the two components may depart from each other to a much greater extent.

(e) LIFT and TIPHON gateways implement the softswitch model in a different way. While in LIFT the combination of an SG and a GC is a softswitch, much like a SG and an MGC are in TIPHON, the internal balance of functions specified by the two frameworks is not the same, as already explained.

The comparison above reflects a different philosophy regarding the nature and the role of a gateway inside an IPTN of any size: namely, the LIFT gateway is closer to the intelligent telephony switch model, a device that handles signalling and traffic in a reliable, scalable, QoS-friendly and feature-rich manner. In other words, the LIFT model approaches the gateway from a *generalised* view, under which even stand-alone signalling and media entities, such as signalling servers and transcoding processors, are classified as SGs or MGs, i.e., as parts of a gateway (perhaps with the addition of a thin software layer, to act as their GC). This allows all application layer VoIP devices to participate in call routing decisions, the same way a gateway does, while maintaining the IP bias of the framework.

4.4.4 Call Routers (CRs)

Call routers are responsible for distributing the information necessary to achieve application-layer connectivity within an IPTN. They maintain databases with local or global reachability information (depending on implementation) that allow any two terminals within an IPTN to establish communication. Contrary to IP routers, CRs do not participate in actual packet forwarding (a task left for layer 3), but they may be assigned with route decision duties (again, depending on implementation).

4.4.5 IP Telephony Administrative Domains (ITADs)

ITADs are mutually disjoint, application-layer domains that partition the entire space of terminals, gateways and call routers of an IPTN. Connectivity among ITADs is advertised by CRs and implemented by (border) gateways (CAPs), at the application layer. Direct connectivity is also possible (e.g., between two gateways that act as SIP Proxy Servers), but provider policies will probably disallow it.

Every ITAD is constructed by at least one call router, zero or more gateways and zero or more terminals. Other devices (e.g., signalling servers or softswitches) can be present but each can be considered as a special case of a gateway, as seen above. The exact formation and connectivity of an ITAD depends on technical and administrative policies applied by the entity (e.g., an ITSP) operating the ITAD.

4.4.6 Organisation

The performance of any VoIP network is significantly dependent upon the topology of its hosts, due to the impact the latter has on call routing (speed and convergence) and data communication (including delay, path stability and packet loss).

Routability mandates a structured organisation of the address space. Even in plain vanilla IP networks, there is a 3-level hierarchy, composed of a top (indirect) level, the Autonomous System (AS), and 2 more direct levels (Network, Host), as acknowledged by the IETF call routing protocol, TRIP [314]. This argument is reinforced by taking into account the use of Variable Length Subnet Masks, which makes the IP address space a 4-level (or, at least, 3.5-level) hierarchy, consisting of AS, Network, Subnet and Host levels. At any rate, LIFT models a large-scale architecture, so address routability is mandatory; higher (i.e., more than 4-level) and more explicit hierarchies are allowed, particularly if source routing is to be applied.

4.5 Call Routing

The main function of IP Telephony call routing protocols at the application layer is advertising and computing gateway reachability information, so that an end-to-end call between any pair (or larger group) of terminals can be established, subject to prevailing policies. At the network layer, call routing is handled by appropriate TCP/IP protocols (e.g., OSPF [275] and BGP-4 [306]), so no special device (call router) is required.

4.5.1 Functional Requirements

The core functionality of any LIFT-compliant call routing scheme must include:

(a) A method by which the ITAD organisation is defined, i.e., how hosts are grouped into ITADs and the inter-ITAD connections. This includes neighbour discovery, for (or after) the formation of each ITAD.

(b) A routable addressing scheme for identifying nodes in the network (including any necessary address lookups and mappings).

(c) A mechanism for route advertisement (i.e., which call routes are advertised and how these routes join neighbouring call routers), for both intra-domain and inter-domain scope.

(d) An algorithm for calculating end-to-end paths (routes), given a source node address, a destination node address and a set of constraints. Path calculation can be performed on a source routing or hop-by-hop routing basis, for both intra-domain and inter-domain scope.

(e) A balanced call routing process, equally capable of supporting heterogeneous (SCN-to-IP and IP-to-SCN), as well as homogeneous (SCN-to-SCN and IP-to-IP) end-to-end calls.

4.5.2 Local and Global Scope Call Routing

Administrative decisions within a large-scale infrastructure such as the ISTN cannot be assigned to a single authority, so a number of organisations (telcos or other) will be responsible for specific areas each, as defined by topological, geographical or other criteria. This partitioning of the network dictates two variations of call routing:

(a) Local (intra-domain, intra-ITAD): Within the same ITAD, with relatively homogeneous administrative policies, call routers can learn about their peer hosts either by static configuration, or through some dynamic means (e.g., periodic route flooding or a special protocol such as SLP [141]).

(b) Global (inter-domain, inter-ITAD): Across different ITADs, call routers need a mechanism for the exchange and synchronisation of routing information, so that they can provide end-to-end paths for call packets. The potentially diverse nature of this information, in conjunction with the need for constraint-based path selection, render conventional directory services and address resolution-based protocols inefficient for this purpose [316].

Because policy-based networking decisions are traditionally hidden within/partitioned in administrative domains (e.g., ISP-controlled), emphasis is primarily given on the inter-domain area. This preference should not disregard the importance of the intra-domain part of the problem, which is also significant, due, among other things, to size restrictions (e.g., large telco-controlled ITADs) and the better chance for optimisations within the borders of an ITAD.

4.5.3 Connectivity

All terminals in a LIFT-compliant IPTN are made accessible via IP, either directly (e.g., if the terminal is a standalone workstation) or indirectly (i.e., if it is a GSTN device, like a telephone, or a computer behind a firewall), whereby a suitable gateway offers it the necessary IP services. All gateways have IP connectivity, and many of them (the ones residing at the boundaries between GSTN and IP networks) GSTN connectivity. All call routers enjoy IP connectivity, too.

From a functional perspective, LIFT exploits the benefits of the Internet Protocol in providing "always on", application layer interconnection, for all entities (terminals, gateways and call routers) involved in VoIP calls. This functionality is implemented via appropriate VoIP signalling and media protocols. Descending the protocol stack, transport and network layer communication is offered by the core IP stack, whereas data link and physical layer details are masked by the ability of the protocol to operate over a wide variety of lower layer technologies. This ability allows large-scale IP Telephony services to be equally offered over dumb or smart voice terminals, wired or wireless connections and PCs, laptops or PDAs.

By design, LIFT decouples IP Telephony call routing from the heterogeneity of the underlying transport networks and assumes a fully (and seamlessly) connected topology instead. In reality, this separation is not strict. Control, signalling, routing and other data can be interleaved in arbitrary ways, as they are all simply binary flows. Furthermore, the end nodes (terminals) do not always enjoy direct connectivity: every call involving two such terminals may have to cross one or more gateways, for a variety of reasons (e.g., gateway capabilities, potential signalling/media conversions, administrative policies, pricing, time of day, or user location). For the same reason, the inbound and outbound paths of a call may not be identical, even at the application layer.

Conceptually, a LIFT-compliant IPTN is quite similar to the multicast overlay model of the MBONE [103]. All participating entities (call routers, gateways and terminals) need to be routed, at the application layer, independently of the underlying network, the topology of which is perhaps taken into account when defining the routing parameters (constraints), but not otherwise during the call routing protocol operation.

4.5.4 Path Decomposition

The end-to-end application layer path for every call in an IPTN can be functionally decomposed into three distinct sub-paths, as demonstrated in Figure 4.1.

4.5.4.1 Access Path (A-Path)

The intra-domain path between a terminal and the gateway that offers it access to the IPTN (i.e., the respective NAP of the terminal) is the "access path" (A-PATH). This is, by definition, topology and call-related.

The A-PATH can be anything from "null" (when the terminal does not need a NAP), to (more usually) a direct connection or a more complex application layer path, consisting of a number of hops.

4.5.4.2 Border Path (B-Path)

The intra-domain path between the NAP of a terminal and the "exit point" of its ITAD (i.e., the respective CAP of the terminal) is the "border path" (B-PATH). This is, by definition, topology and call-related.

The B-PATH can be anything from "null" (when the NAP is also a CAP), to a direct connection or a more complex application layer path, comprising of a number of hops, from the terminals NAP to the CAP (as in Figure 4.1, ITAD3).

4.5.4.3 Carrier Path (C-Path)

The inter-domain path between any two border gateways (CAPs), is the "carrier path" (C-PATH). For the purposes of a call, a C-PATH extends from the CAP serving the source terminal to the CAP serving the destination terminal. This is, by definition, topology and call-related.

The C-PATH can be anything from "null" (when the caller and the callee are served by the same ITAD), to a complex application layer path, crossing multiple ITADs on the way from source to destination (and vice versa).

4.5.4.4 Path Combinations

The A-PATH, B-PATH and C-PATH in any call are adjacent but disjoint portions of the end-to-end call path. Depending on the call setup scenario, combinations of those may be defined, so that a larger path is formed. An AB-PATH, for instance, is the intra-domain path from a terminal to its Carrier Access Point, built from the concatenation of an A-PATH and a B-PATH.

4.5.5 Path Selection Parameters (Constraints)

Routing parameters (attributes or metrics) are defined by administrative means (e.g., by ITAD operators), distributed dynamically (by call routing protocols) and applied locally during path calculation. There are several constraints that can be used in a LIFT-compliant IPTN, all focused on the application layer protagonist device – i.e., the gateway. Such parameters are:

(a) **Capacity**: A characterisation of the processing power of the gateway, referring to features such as CPU power, buffer sizes, number of ports, or the Busy Hour Call Attempts (BHCA) it can support.

(b) **Bandwidth**: For each node, the minimum of its own bandwidth (i.e., the output rate of the node, restricted for example by its CPU and its network interface) and the link bandwidth.

(c) **Delay**: The sum of the processing delay at the gateway and the total link delay.

(d) **Cost**: The monetary cost for a call using the resources of the gateway.

(e) **Signalling capabilities**: Signalling protocols supported by the gateway, either stand-alone, or combined (for translation purposes).

(f) **Media capabilities**: CODECs and media formats supported by the gateway.

(g) **Intelligent Network features**: Standard or advanced telephony services (e.g., call waiting/hold/forward/transfer) implemented in the gateway.

The plethora of routing parameters can, in theory, allow for a very user-friendly network, in which customers select their service based on criteria such as functionality and cost. In practice, however, increasing the number of constraints progressively converts path calculation to an NP-hard problem, so alternative strategies must be applied. Three solutions are proposed for tackling this problem:

(a) *Aggregation.* Replacement of all routing parameters propagated by a single one (a "superconstraint"), which is calculated from the others (e.g., using a weighted sum equation).

(b) *Classification.* Combination of constraints with common characteristics into groups, using them instead of individual constraints. For instance, from the above set of attributes, capacity, bandwidth and delay could be grouped as "scalability", whereas all media capabilities could also be grouped into one routing parameter.

(c) *Simplification.* Restriction of the set of constraints by preferring only a few "basic" ones and ignoring the rest. This is the most pragmatic and easy to implement strategy, allowing for potentially shorter path calculation times and less complex routing topologies.

The constraint selection method directly affects end-to-end path calculation and, through that, the blocking characteristics of a network. In traditional telephony, because of the circuit switching operation, this problem is solved using backtracking ("crankback") techniques [9]. A similar mechanism is also needed for packet switching networks, which can also exhibit blocking characteristics if mechanisms like admission control or resource reservation are applied, although these can be minimised through the choice of appropriate routing constraints. In any case, LIFT assumes that, if implemented, blocking resolution will happen at the very "last minute", i.e., after path selection, call setup and all subsequent signalling operations (e.g., QoS provision) have been completed.

4.5.6 Path Calculation

The partitioning of the end-to-end call path into three distinct portions (A-PATH, B-PATH and C-PATH) allows different mechanisms to be used for path calculation. A terminal, for instance, can be configured with its A-PATH and discover its B-PATH (via its NAP) using SLP [141], LDAP [384] or even an existing IP link state routing protocol like OSPF [275]. Paths outside the terminal's ITAD will often be much less static, so the main target of the call routing mechanism will be the calculation of C-PATHs for every call, which can be executed either on an end-to-end or on a per-hop basis, using source routing or hop-by-hop routing, respectively.

LIFT adds to the complexity of the path calculation problem by making the assumption that a (small, but possibly higher than one) number of gateways will need to be involved in an average IP Telephony call, and these gateways will likely advertise inconsistent routing parameters for path weighting. This fact, along with the connection-oriented operation of IP Telephony at the application layer, suggests that some sort of "permanent" sequence of gateways could be most efficient for the media path of a call (the signalling path is more-or-less stabilised, although different, as seen in Chapter 3). This is because such a media path would allow faster adaptation to changes in end-user requirements (e.g., on-the-fly CODEC alterations) and could also be helpful in issues specific to real-time media transmission, such as packet re-ordering and delay variation. In addition, it would be more QoS-friendly, particularly if resource reservation is used for the media path.

The above "route pinning" philosophy is compatible with routing mechanisms found in RSVP [43] and MPLS [308], [310], as well as in PNNI [11]. However, in a LIFT-compliant network it would be applied at the application layer and for a much smaller number of nodes (i.e., the few gateways in the end-to-end path, instead of all the network layer routers), so it could be called "semi-pinning".

4.6 Transport Layer Issues

A variety of transport layer issues are applicable to a generic architectural framework like LIFT, but their resolution will be implementation-dependent. Special consideration, though, is needed for reliability and multicasting.

4.6.1 Reliability

LIFT assumes that transport layer reliability will vary according to application layer requirements: connection-oriented protocols (like TCP and SCTP) will continue to be more suitable for control information (including signalling) and non-interactive media, whereas connectionless ones (like UDP) will be used selectively for the control plane and almost always for interactive media.

However, call routing in a LIFT network has a set of minimum reliability requirements. Each call router is assumed to be able to open a "direct" and reliable connection (e.g., a TCP socket) to exchange routing information with any other call router to which it is logically (i.e., physically and administratively) connected. Call routers should also be able to receive and transmit information on a point-to-multipoint basis (e.g., via multicasting), for flooding purposes, which means they should be able to operate over an unreliable connectionless protocol (e.g., UDP).

4.6.2 Multicasting

Multi-constraint multicast call routing is bound to be prohibitively costly in any terms [257], [392]. LIFT assumes that multicast IP telephony will involve only small to medium-size IP multicast-capable groups, and that large-scale conferencing constitutes a separate problem area that merits further investigation, beyond the scope of a generic architectural framework.

Within such an environment, terminals wanting to participate in a multicast call would receive packets in a centralised fashion, through a media server, of which the address they will know by some other means (e.g., SAP [152]). Then, the source, as well as each terminal wanting to join, will execute "individual" call routing path calculations with the media server as the destination, but with aggregation of QoS requirements (e.g., in a RSVP-like fashion) towards the media server, so that the multicast distribution can be performed afterwards.

4.7 Network Layer Issues

Most network layer issues applicable on LIFT are expected to be resolved according to implementation, but certain provisions are necessary for addressing and QoS.

4.7.1 Addressing

Every IP Telephony host is characterised by its routable address, which is defined as an IP (IPv4 or IPv6) address ("partial" address) for call routers and a <E.164-address, IP-address> pair for callable devices, i.e. terminals and gateways ("full" address). This address pair will be statically configured with every terminal and gateway, or occur as the output of a mapping function, for improved flexibility. For example, the E.164 address of a workstation could be discovered as a result of a directory lookup, with its IP address as input, or using a special protocol like DHCP [97] for this purpose. Either of the two identifiers of the address pair is adequate for locating the host in the network, but they are both necessary for usability purposes.

Addresses (partial or full) must be assigned by a central authority and deployed through ITAD-specific administrative policies. Like with TRIP, virtual addresses (e.g., 0800 numbers) are not propagated, therefore a separate mechanism will be needed for supporting non-geographic calls. Route advertisements will always contain full addresses. All legacy devices will be allocated with an IP address. Multiple addresses are allowed for all hosts. Address depletion will be countered with the full adoption of IPv6 in the near future.

4.7.2 Quality of Service

In LIFT-compliant networks, if QoS is implemented, the routing decision will be executed in two successive steps:

(a) Discovery of a path from source to destination that will satisfy the constraints set.

(b) Provisioning, allocating or reserving the necessary resources end-to-end.

As discussed already, the implications of path discovery are closely related to the constraints advertised. If QoS is carefully taken into account in the selection of the routing parameters, blocking probabilities should be significantly reduced.

The resource reservation, whenever necessary, can happen either after the call is set up at the signalling level and before the media starts flowing, or by attempting reservations simultaneously with setup. The latter approach is rather more difficult, because IP QoS mechanisms (e.g., RSVP) operate better on paths between end points, rather than on a segmented basis that can be combined with setup.

4.8 Security Issues

The architecture-related security considerations examined by the TRIP Framework [316] are also acknowledged by LIFT. For VoIP-specific security issues, the techniques and the solutions analysed in Chapter 3 are generally applicable. Additional security-related issues extend beyond the scope of this framework.

4.9 Billing

Billing for IP services remains a largely unexplored area, which, however, is essential for the establishment of a global scale ISTN [82]. LIFT assumes that IP Telephony providers will apply a variety of policies to charge according to special agreements (SLAs) with end users, probably based on both traffic volume and time, along the lines of Chapter 3; however, the implementation of the framework does not depend on particular billing schemes.

4.10 Implementation

As a generic framework, LIFT does not specify implementation details. However, the following characteristics are strongly recommended for every compliant ITPN:

(a) *Standardisation.* All software implementations should comply with established industry standards, to maximise interoperability of VoIP equipment and networks.

(b) *Openness.* To facilitate deployment, applications should be made fully available to the public, down to the source code level, if possible.

(c) *Portability.* Maximum effort should be applied towards the direction of developing applications that are portable across at least the most popular combinations of hardware and operating systems.

The above principles are applied on the LIFT-compliant implementations presented and analysed in Chapters 5, 6 and 7.

4.11 Evaluation

LIFT describes an intentionally generic architecture, so in practice it can only be evaluated comparatively to other similar specifications. For this purpose, the framework is analysed in terms of *innovation* (i.e., differentiation from existing standards) and *applicability* (i.e., implementation feasibility of its main components).

Regarding the first case, as already discussed, a number of frameworks define, partially or fully, the architecture of a large-scale IPTN. All these models are more or less converging to a single architecture, with which LIFT has several similarities, but also a number of differences, in the following areas:

(a) **IP Telephony User Agent**: LIFT considers the IPTUA as an indispensable part of an IPTN. Furthermore, it mandates that all such applications be capable of joining, participating in and leaving conference calls without relying on external means for doing so; such a requirement implies that, in addition to media handling, audio tools must also have full signalling and other control capabilities, as already discussed.

(b) **Gateway architecture**: LIFT specifies a more generic type of gateway, which encompasses all existing signalling and media components into one entity that can be scaled down or up to any set of features (from a lightweight signalling or media server, to a powerful softswitch), as explained above.

(c) **Number of gateways per call**: LIFT allows a multiplicity of gateways to be involved in an IP Telephony call at both the signalling and the media path, although such an approach should be avoided or restricted whenever possible, for performance reasons.

(d) **Call router functionality**: LIFT defines call routers as devices that handle application layer call routing information (including, potentially, decision making), but are not involved with the actual forwarding of call packets.

(e) **Bidirectional call routing**: LIFT expects that call routing will also be needed for the SCN-to-IP direction, so the aim of the call routing protocol should not be restricted to discovering a gateway to complete an IP-to-SCN call, as is in TRIP.

(f) **Call path definition**: LIFT assumes a more static ("semi-pinned") view of a call path at the application layer, including specific access points (gateways) on a per call basis, for all types of IP Telephony terminals.

(g) **Stabilisation of both the signalling path and the media path**: LIFT perceives disjoint signalling and media paths like the other frameworks, but increases the stability of the media path by introducing the semi-pinned path concept and by allowing source routing for path calculations.

(h) **ITAD scope**: LIFT incorporates all IP Telephony devices (terminals) in an ITAD, as opposed to being restricted to provider-related resources (e.g., location servers and gateways, as in TRIP [316]).

(i) **Level of abstraction**: LIFT is generic enough to be independent of particular control and data plane protocols, while at the same time specific enough to define an architecture where large-scale IP Telephony is possible.

The extent to which the above differences affect the performance of a LIFT-compliant IPTN, compared to IPTNs based on other VoIP frameworks, will depend on the percentage of implementation of the LIFT guidelines.

For the assessment of the applicability of LIFT, working implementations of the three representative components identified in Chapter 1 (i.e., the IPTUA, the gateway and the call routing protocol) are presented and evaluated in Chapters 5-7. An additional indication about the practicality of the framework can be derived from a comparison of the other existing IP Telephony frameworks with the widely deployed and field-proven GSTN architectural model, which is based on SS7 [26], [198]. Indeed, SS7 networks, responsible for handling the control traffic of the digital telephony network, are packet switched, they assume different signalling and media paths, they implement scalability via distributed switching logic and they aim at establishing global connectivity; LIFT, with its generalised view of a gateway and its other GSTN-like characteristics already explained, is closer to the SS7 operational model than any other IP Telephony framework currently specified.

4.12 Conclusion

Existing IP Telephony architectural frameworks tend to focus mainly on the IP side or primarily on the GSTN side of the communication, while overlooking a vast number of conventional telephone users, the smooth integration of whom in a future, universal VoIP network will be crucial in the success of the technology at a global scale. Ignoring this fact probably constitutes one of the main reasons why IP Telephony is still restricted in the backbone, despite its unanimous adoption by both the academic and the commercial community.

This chapter has described a new approach, LIFT, which addresses the larger picture and sets the guidelines necessary for a smooth transition to the global ISTN. LIFT assumes that such a transition can only take place by exploiting the benefits of the field-proven GSTN architecture, isolated elements of which are also progressively adopted by the other frameworks, as they mature and expand to large-scale implementations. LIFT further differentiates itself from other similar proposals in a number of areas, including the IP Telephony user agent, the gateway, and the call routing mechanism, novel views to all of which are presented in detail and evaluated in the following three chapters.

CHAPTER 5

An IP Telephony User Agent

5.1 Overview

An IP Telephony User Agent (audio tool) is a software application capable of exploiting the computer hardware on which it runs for both placing and accepting voice calls on behalf of an end user, thus also being referred to as a *softphone*. To achieve this, it implements standard VoIP control (signalling) and data (traffic) protocols and provides a User Interface (UI) through which the necessary functionality is accessible.

This chapter presents a novel audio tool architecture and the design, implementation and evaluation of a prototype application (the Configurable Audio Tool, CAT) that follows it, while also complying with the functional requirements set by LIFT in Chapter 4 and exploiting a significant amount of research experience derived from a 3-year involvement with the Robust Audio Tool (RAT) project and its team in UCL-CS. This way, the first of the three representative areas identified in Chapter 1 is investigated and the corresponding part of the hypothesis is accordingly supported.

5.2 Related Work

A multitude of IP Telephony User Agents have appeared since the ISI Voice Terminal (vt) and its successor, the LNBL Visual Audio Tool (vat) [U58], were first developed, in 1991 [257]. As discussed in Chapter 2, from the 1990s and onwards these can roughly be considered as "second generation" and are further classifiable according to a dual taxonomy: applications that originated from the academic world (e.g., vat [U58], NeVoT [330], RAT [153] and FreePhone [37], [38]) were oriented towards the MBONE [103], [252], while those coming from the private sector (companies or individuals) followed more proprietary routes (e.g., Internet Phone [U59] and Speak Freely [U49]). What both categories had in common was their focus on the media path, mainly due to the indifference for, experimentation with, or slow deployment of standardised signalling protocols. Significant advances were thus made in the media field, particularly in CODEC support, error concealment and correction, playout buffering, synchronisation and mixing, while the core IP Telephony signalling and media protocols (including H.323, SIP and RTP) were being standardised [286]. The UCL-CS Robust Audio Tool has been at the forefront of these developments from the start and, by evolving from a simple packet voice client to a sophisticated audio research platform [154], [155], [163], it has contributed significantly to the progress of the entire IP Telephony User Agent field. In fact, to some extent RAT is still used as a reference point [143], although its development has been practically frozen as of late 2003 and only patch work is being done ever since [U43].

Due to the volatile nature of software development, many of the academic audio tools and (for obvious reasons of privacy) almost all the ones coming from the private sector do not document well their architecture, even in the publications discussing a tool itself. The usual practice is analysing the structure of the media path by focusing on the algorithms used for solving the most important operational issues (e.g., compression, error concealment or playout) and not including any clear API definitions [37], [38], [153], [330]. The emergence of a third generation (mainly softphones) in large numbers has practically converted audio tools into commodity applications that can be developed fairly easily, either from scratch, or, even faster, by using standardised third party libraries, like the Java Media Framework (JMF) [U51] or JVOIPLIB [247]. As a result, the likelihood of new audio tool architectures being created or published henceforth has been significantly reduced.

That being said, interesting relevant work has been recently documented. This ranges from implementation guidelines on RTP-based audio and video applications [237], [286], to the development and performance analysis of new IP Telephony User Agents, components or techniques [67], [143], [212], [231], [248], [279], [320], [340]. However, a specific architecture and implementation presented to the degree of detail used for CAT herein only appears in [234], where multiple signalling protocols are incorporated in a single IPTUA, but just one media engine is used. By contrast, CAT allows the use of different signalling protocols and pluggable media engines, three of which are evaluated here, and follows the LIFT implementation guidelines set out in Chapter 4.

The work on CAT presented in the rest of this chapter and on the MG of the gateway discussed in Chapter 6, draws considerably from the participation in the Networked Multimedia Group of UCL-CS and the RAT project, as well as from experimenting with some other public domain audio tools, like vat [U58] and Speak Freely [U49], during the second half of the 1990s. More specific details about the technologies discussed in this section and the rest of the chapter can be found in the corresponding parts of the literature review, presented in Chapters 2 and 3.

5.3 Architecture and Design

The architecture of the IP Telephony User Agent proposed herein, analyses the tool into a number of basic components, of which the ones related to signalling and media encompass the core VoIP functionality and, therefore, are of primary importance. Designing the application in a modular way also facilitates portability, which is a significant benefit in the effort for large-scale IP Telephony deployment, given the abundance of computing platforms in operation. The inevitable performance penalty incurred by modular designs [74] can be offset by careful implementation [233] and also by the sheer processing power found even in lower-end computing equipment, as explained in the following sections.

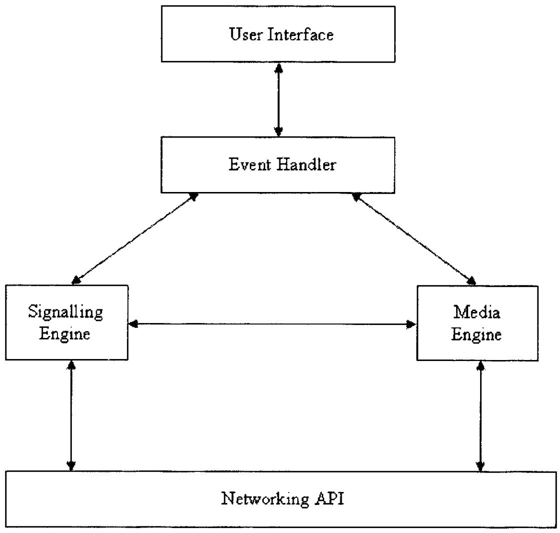


Figure 5.1: Top-level architecture of an audio tool

5.3.1 Main Components of an IP Telephony User Agent

A functional analysis of the internals of an audio tool identifies a number of high-level modules, presented in Figure 5.1 from a cross-platform application-layer view, i.e. independently of activities related to the Operating System (OS):

- **User Interface**: This is the component responsible for drawing and updating the audio tool screen in real time, based on information received and transmitted; it also allows the user to configure the application, according to the available set of options (Property Set), as seen in Table 5.1. Most commonly, a Graphical UI (GUI) is employed, but a command-line UI or a GUI with a command console, are equally suitable.
- **Event Handler**: The existence of numerous different functions within the IP Telephony User Agent implies the need for a means of internal communication; to assist modularity, this communication should be message-based, so a special process, the event handler, is

needed. The event handler will maintain one or more queues where messages from other components are deposited and dispatched according to the application's scheduling policy. Part of these messages will be directed to the UI, updating information such as speech volume, conference participant information, audio device status and traffic statistics.

- **Signalling Engine**: This component is primarily responsible for establishing (placing or accepting) a call - in other words, it is comprised of the set of functions that implement the VoIP signalling protocol(s) (usually H.323 or SIP, but possibly additional ones, such as H.248/MEGACO). Other functions may include call logging, processing of control and management information, and communication with extra signalling entities such as a directory server.
- **Media Engine**: Handling incoming and outgoing voice traffic is the main role of this component. It implements the standard VoIP media encapsulation protocol, RTP, and all stages of the voice pipeline inside the audio tool, including, among other features, encoding, silence suppression, packet processing, error handling, synchronisation and transmission or reception through the Networking API.
- **Networking API**: This component is based on the standard set of TCP/IP system calls (accessible via the programming language in which the audio tool is written), which are expanded by proprietary functions ("glue logic"), if necessary. (Support for other network technologies, beyond IP, requires extending this API.) The Networking API is used by the Signalling Engine and the Media Engine for exchanging information over TCP or UDP sockets; this is done independently, so no need for a centralised packet dispatcher exists.

Property	API
open_packet_interface()	PACAPI
send_packet()	PACAPI
receive_packet()	PACAPI
close_packet_interface()	PACAPI
start_media_engine()	CAPI
get_media_engine_properties()	CAPI
set_media_engine_properties()	CAPI
stop_media_engine()	CAPI

Table 5.1: The CAT property set

5.3.2 Core Functionality

The core VoIP-related functionality of the audio tool is implemented by two components, the Signalling Engine and the Media Engine, which interact directly with each other, as well as with the Event Handler and with the Networking API, as depicted on Figure 5.1. A more detailed view of this interaction can be seen on Figure 5.2, which presents the proposed IP Telephony User Agent architecture. (A microphone and a speaker are shown alongside main components; internal control channels appear as thick lines.)

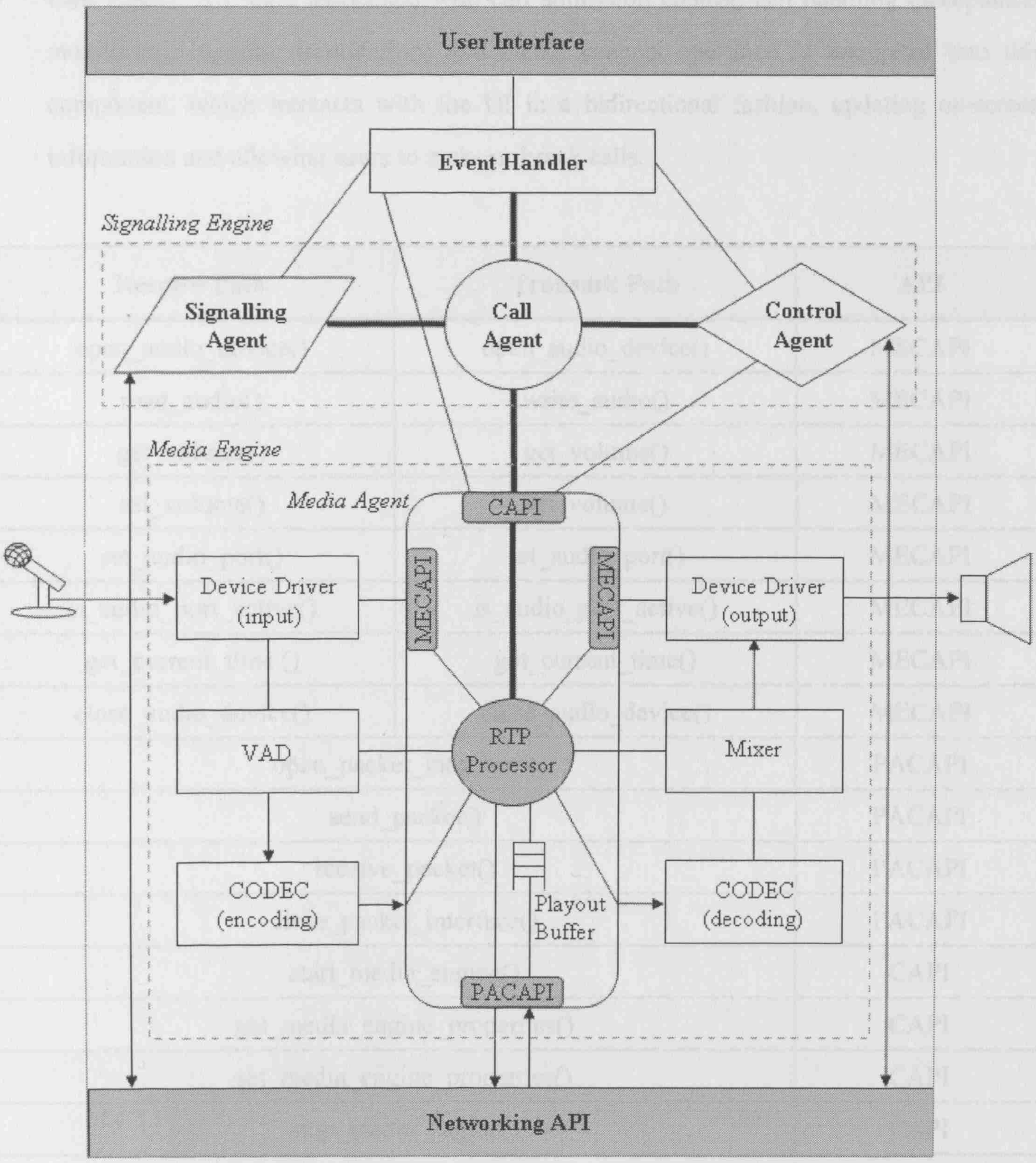


Figure 5.2: Generic Architecture of an IP Telephony User Agent

5.3.2.1 Signalling Engine

The Signalling Engine of the audio tool comprises of a number of components, as shown in Figure 5.2:

- **Signalling Agent**: This function implements the VoIP signalling protocol(s) used by the application for placing or accepting calls, according to user instructions.
- **Control Agent**: This entity accumulates a number of control-related activities, which are largely implementation-dependent. For instance, it may incorporate a directory services client, a network management module, an instance messaging function, or some other (standard or proprietary) mechanism, which may interact with the UI.
- **Call Agent**: All logic associated with call admission control, call handling (acceptance, monitoring, logging, termination) and media channel operation is integrated into this component, which interacts with the UI in a bidirectional fashion, updating on-screen information and allowing users to make or break calls.

Receive Path	Transmit Path	API
open_audio_device()	open_audio_device()	MECAPI
read_audio()	write_audio()	MECAPI
get_volume()	get_volume()	MECAPI
set_volume()	set_volume()	MECAPI
set_audio_port()	set_audio_port()	MECAPI
is_audio_port_active()	is_audio_port_active()	MECAPI
get_current_time ()	get_current_time()	MECAPI
close_audio_device()	close_audio_device()	MECAPI
open_packet_interface()		PACAPI
send_packet()		PACAPI
receive_packet()		PACAPI
close_packet_interface()		PACAPI
start_media_engine()		CAPI
get_media_engine_properties()		CAPI
set_media_engine_properties()		CAPI
stop_media_engine()		CAPI

Table 5.2: Main functions of the Media Agent APIs

5.3.2.2 Media Engine

The Media Engine of the audio tool is also visualised as a collection of smaller functional blocks, as shown in Figure 5.2:

- **Media Agent**: The Media Agent is the core of the Media Engine, responsible, first of all, for handling the audio device (through a special set of functions, the Media Control API, "MECAPI"), after being instructed so by the Call Agent (similarly, through the Control API, "CAPI" - not to be confused with the Common ISDN API). It also updates the UI with media-related information (e.g., speech volume) in an event-driven fashion (through the CAPI), performs packetisation (prior to transmission) and depacketisation (after reception), and maintains timing dependencies (for synchronisation). On the receive path, it additionally manages the playout buffer, dynamically adjusting it to application activity (including load analysis of the transmit path) and prevailing network conditions (via information received from the Control Agent). The interaction of the Media Agent with the Networking API happens through the Packet API ("PACAPI"). The main functions of the CAPI, the MECAPI and the PACAPI are listed in Table 5.2.
- **RTP Processor**: This component implements the RTP-related functionality of the Media Agent. Its duties include creating and handling data (RTP) and control (RTCP) media packets, maintaining and distributing session information, coordinating with other parts of the media engine (e.g., the CODECs) and preserving the timing dependencies found in the media stream.
- **Device Driver**: Digitised voice is read from the audio device (i.e., the computer's sound circuit) for transmission and written to it for reception; the audio device hardware is responsible for analogue to digital (A/D) and digital-to-analogue (D/A) conversion so that the attached audio capture (e.g., microphone) and playback (e.g., a speaker) devices can operate in a humanly sensible manner. The digitised voice stream is exchanged between the audio hardware and the audio tool application through calls to the MECAPI.
- **Voice Activity Detection (VAD)**: The decision whether a digital voice stream should be transmitted, or ignored as silence, belongs to this function and is usually based on energy level measurements. Whenever silence is detected (i.e., when there is no signal, or the signal strength is below a preset threshold), transmission can stop altogether, or, to maintain the impression of an open channel in both ends, comfort noise can be generated and sent once, or at sparse periodic intervals.
- **CODEC**: This component encodes voice during transmission (prior to packetisation) and

decodes it during reception (after depacketisation). The two modules (encoding/decoding) are usually part of the same implementation, as the process is roughly symmetric.

- **Playout Buffer**: This is a simple data structure responsible for storing a small number of incoming packets so that network-related problems (mainly packet delay, jitter, duplication, re-sequencing and loss) can be countered by the Media Agent. Playout buffer management strategies can vary from simple static techniques to sophisticated algorithms that adapt dynamically to the behaviour of the network, as discussed in Chapter 2.
- **Mixer**: On the reception path, combining ("mixing") voice packets from different sources, such as during a conference call, is necessary whenever this operation has not been carried out by an external entity; this is precisely the task of the audio tool's Mixer module.

It is important to point out that the above presentation is a logical one; depending on implementation (e.g., programming language and particular algorithms used), some components may be co-located and internal order and connectivity among components may vary.

5.3.3 Portability

Due to the variety of existing OS and hardware combinations, developing a cross-platform VoIP client application requires careful design and considerable effort in two fronts, the programming environment and the intended functionality. While advances in popular high-level programming languages (like C, C++ and Java) have led to the creation of portable libraries, standardised APIs and less incompatible binaries, universal implementation of the same functionality remains a sizeable challenge, despite the alleviation of older problems such as the lack of processing power or the existence of intricate sound devices (especially in some Unix systems).

A properly designed audio tool should use well-defined APIs that allow the functional expansion of both the Signalling Engine and the Media Engine objects (e.g., to support more protocols). However, as indicated by Figure 5.2, the most complicated part in that respect is the Media Engine. This happens because the other audio tool modules are either simpler (e.g., the UI) or incorporate standard protocols (e.g., the Signalling Agent); the Media Engine, on the other hand, beyond its standard (or quasi-standard) protocols (such as RTP), components (e.g., CODECs) and algorithms (e.g., for playout buffer management and mixing), always requires significant amounts of application-dependent code to complete its implementation.

Despite these difficulties, the portability of the Media Engine can be significantly improved using a modular, object-oriented design for its core, the Media Agent, in order to isolate it from the external functional modules through appropriate APIs (Figure 5.2 and Figure

5.3). This way, the entire Media Engine is fully encapsulated ("wrapped" inside) a small set of well-defined, easily portable functions.

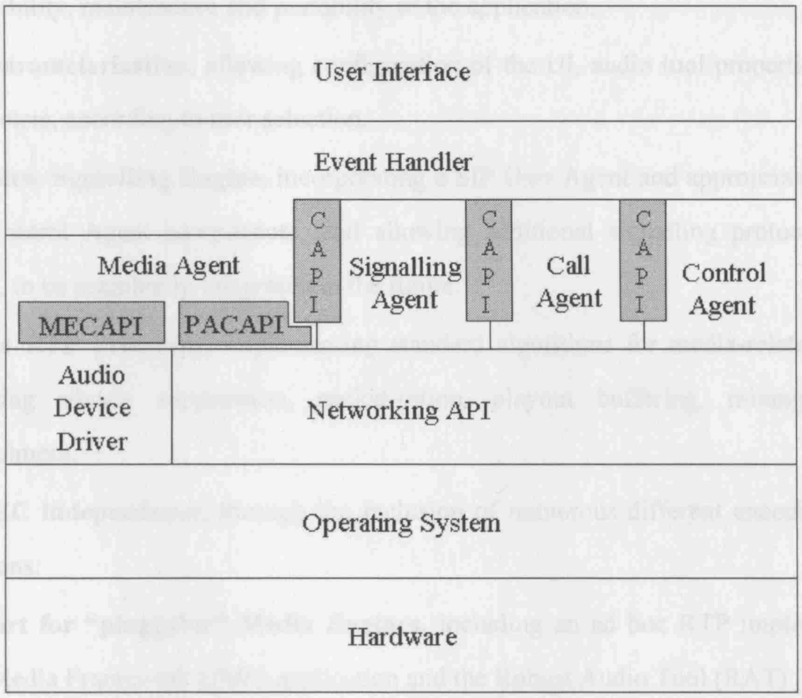


Figure 5.3: The system stack of an IP Telephony User Agent

In fact, if the isolation is complete, the concept of "pluggable" Media Engines is possible, allowing the decomposition of an audio tool, much like VoIP gateways, as discussed in Chapters 3 and 4. This, in turn, is useful in moving complicated functionality outside the audio tool (e.g., to the associated gateway), thus allowing the development of simpler VoIP terminals (e.g., low-end IP Phones). On the other hand, many existing audio tools do not integrate signalling or control functionality and instead rely on external means (i.e., user action or third-party mechanisms) to establish voice calls. Such applications are essentially Media Engines with a User Interface and a Property Set configurable through it, so they can be plugged into an IP Telephony User Agent that follows the above architecture, at varying levels of integration.

5.4 A Research Implementation

An instance of the architecture analysed in the previous section has been implemented, as a proof of concept, into a simple IP Telephony User Agent, the Configurable Audio Tool (CAT), using Java, which has been proven an adequate tool for this purpose [1]. The main characteristics of CAT are:

- **Native object-oriented design** around well-defined function modules, to facilitate extensibility, maintenance and portability of the application.
- **Full parameterisation**, allowing configuration of the UI, audio tool properties and VoIP parameters, according to user selection.
- **Complete Signalling Engine**, incorporating a SIP User Agent and appropriate Call Agent and Control Agent components, and allowing additional signalling protocols, such as H.323, to be seamlessly integrated in the future.
- **Simple RTP Processor**, implementing standard algorithms for media-related functions, including silence suppression, packetisation, playout buffering, mixing and error concealment.
- **CODEC independence**, through the inclusion of numerous different encoding/decoding functions.
- **Support for "pluggable" Media Engines**, including an ad hoc RTP implementation, a Java Media Framework (JMF) application and the Robust Audio Tool (RAT) [154], [163].
- **ATM interface**, allowing media connectivity with ATM-based packet networks.
- **Cross-platform operation**, over the most popular hardware architecture and operating systems.

More details of the above characteristics and their actual implementation into CAT are given in the following sections.

5.4.1 Object-oriented Design

A number of Java classes implement the functionality of CAT, according to the proposed IP Telephony User Agent architecture (Figure 5.2). A simplified UML class diagram of the application is depicted in Figure 5.4, showing the dependencies prevalent among the main software objects.

5.4.2 User Interface

The simple Graphical User Interface of CAT reflects high levels of internal parameterisation to the end user. Apart from the necessary VoIP and application-related configurability, two different versions of the GUI, one for the *AWT* and one for the *Swing* Java libraries, were created (Figure 5.5), while the choice of appearances ("skins") is increased through the different

"Look and Feel" options of the Swing library. The CAT GUI does not offer the polished (and non-standardised) front-end found in most softphones, as emphasis has been placed on the actual architecture and functionality of the audio tool. A similar approach has not obstructed the GSTN with its rather Spartan user interface (i.e., the telephone keypad) to be ubiquitously present and acceptable by the user community; that being said, however, the necessity of a contemporary GUI for increasing the attractiveness and adoptability of a stand-alone IPTUA should not be overlooked.

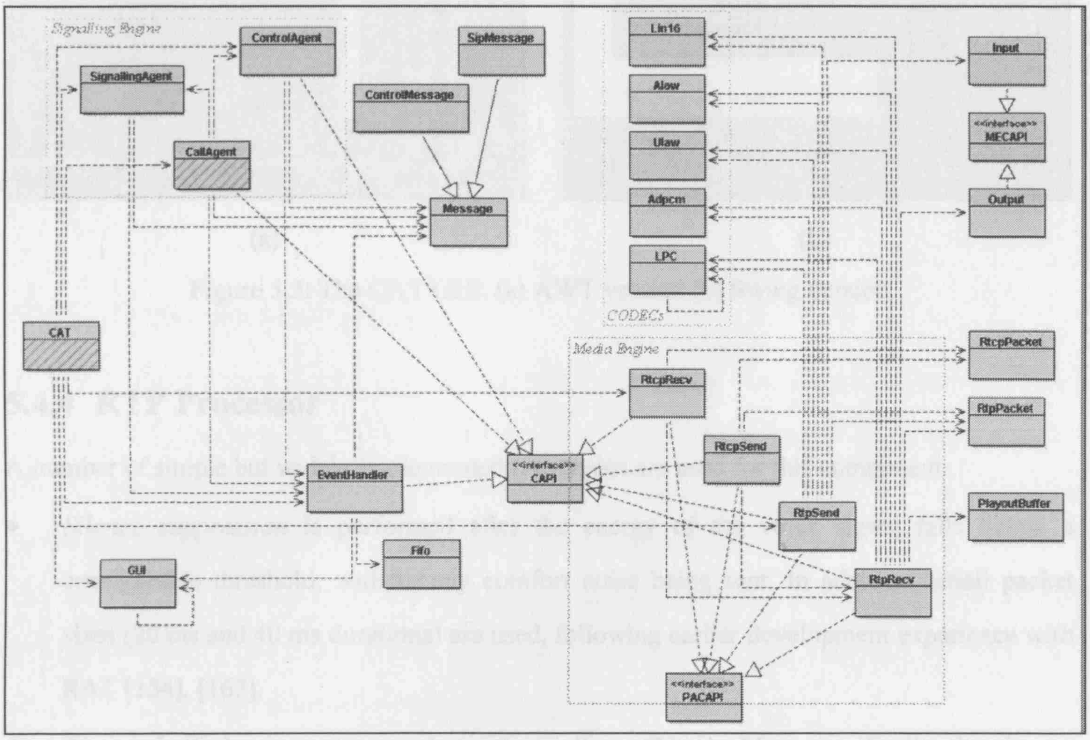
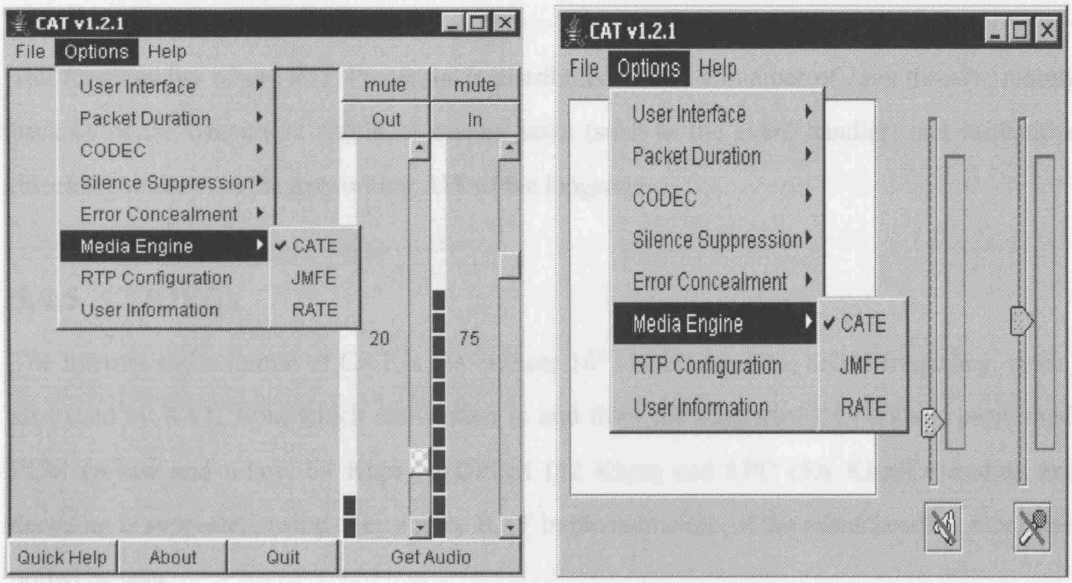


Figure 5.4: Simplified UML class diagram of CAT

5.4.3 Signalling Engine

The main part of the signalling engine is the Signalling Agent, which is a SIP User Agent, i.e., a combination of a SIP User Agent Client (UAC) and a SIP User Agent Server (UAS), so that the tool can both place and receive VoIP calls. SIP over UDP was selected for this implementation. The integration of other signalling protocols is also facilitated by the modular design of the application.

The Call Agent and Control Agent components are thin objects, as they perform very simple functions: call admission control based on destination IP address filtering and processing of internal CAT control messages.



(a) (b)

Figure 5.5: The CAT GUI: (a) AWT version (b) Swing version

5.4.4 RTP Processor

A number of simple but widely implemented techniques are used for this component.

- *Silence suppression* is performed after the energy of the voice signal falls below a configurable threshold, without any comfort noise being sent. In addition, small packet sizes (20 ms and 40 ms durations) are used, following earlier development experience with RAT [154], [163].
- *Playout buffering* is implemented on a "per-talkspurt" basis. More specifically, the playout time of the first packet in each talkspurt is estimated and subsequent packet times are adjusted according to this estimation, following the weighted linear recursive filter algorithm described in [303] and the RTP packet interarrival jitter calculation method [337]. This process continues until the end of the talkspurt, after which it is restarted.
- *Mixing*, whenever necessary, is performed as an averaging function on audio received from related sources. This process is just a summation of individual sample values, performed in the CAT internal audio format, with each sum being divided by the number of contributing packets.
- *Error concealment* is also rudimentary, creating a replacement for every packet lost, either from silence (silence substitution), or from a copy of its correctly received predecessor (packet repetition). Error tolerance of up to 5% (non-consecutive) has been verified from

previous experience of using these methods in RAT [154], [U43] and it was considered enough for the purposes of CAT.

The functionality of the RTP Processor is distributed across a number of Java threads, mainly because of the concurrent nature of certain tasks (such as the event handler) and limitations (blocking sockets) in the networking API of the language.

5.4.5 CODECs

The internal audio format of CAT is the "Linear 16" (16-bit samples, 8KHz frequency, mono) also used by RAT, from which conversion to and from the supported CODECs is performed. PCM (A-law and u-law, 64 Kbps), ADPCM (32 Kbps) and LPC (5.6 Kbps) encoding and decoding is supported, using open source RAT implementations of the corresponding functions, ported in Java.

5.4.6 Pluggable Media Engines

The CAT Media Engine (CATE) is implemented as a number of Java threads that are combined into the RTP Processor, the CODECs and the three Control APIs (Figure 5.2 and Figure 5.3). CATE was developed based on experience and concepts (internal audio format, CODECs, optimisation) coming from earlier participation and contributions to the RAT project [U43]. It can work as a stand-alone audio tool and use an external signalling mechanism (or manual call initiation) to join VoIP conferences. Conversely, media-only audio tools can be "wrapped" inside CAT, like pluggable media engines, as already explained.

Two such applications have been used for this purpose: the Robust Audio Tool (RAT) itself, an advanced VoIP research platform, supporting secure conferencing, multiple channels, CODECs, error concealment methods and operating systems; and an ad hoc Media Engine, created using the Java Media Framework (JMF) API (Appendix 1). The RAT Media Engine (RATE) was based on RAT Version 3 (for toll quality audio, written in C), which consists of two main components, a user interface and an audio engine [163]. A simple CAPI was implemented using the Java Native Interface (JNI), in order to allow CAT to invoke RAT in a "stealth" mode (i.e., without the user interface), set its basic properties (e.g., packet duration and encoding method), get property information (including sound volume and RTCP statistics) and shut it down at call completion.

A similar set of functions was used to control, from within CAT, the JMF-based Media Engine (JMFE), although no native methods were needed, since the JMF library [U51] is "pure

Java". More specifically, following an object-oriented philosophy, JMF models the media lifecycle in a conference, from source to destination, into three broad phases: *input* (capture from a device, file, or the network), *process* (encode, decode, transcode, apply effects) and *output* (present, play). A simplified view of this model is shown in Figure 5.6, where DataSource, Processor and Player objects implement the input, process and output phases, respectively. These are assisted by other objects responsible for audio capture in the appropriate format (AudioFormat, CaptureDevice), setting up RTP session parameters and registering event listeners (Manager), handling transmission and reception of RTP packets over IP (SendStream, ReceiveStream) and eventually replaying at the receiver (Player).

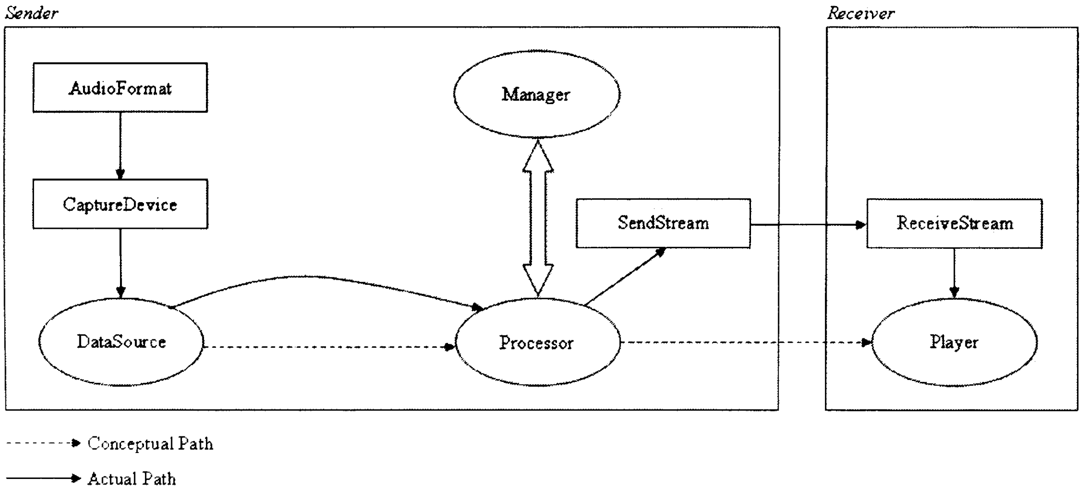


Figure 5.6: A simplified view of the JMF transmission path object model

5.4.7 Java ATM Interface

As a proof of concept, compatibility with ATM networks was built into the CATE, JMFE and RATE Media Engines. In all three cases, call initiation happens manually, as automated signalling between ATM and IP networks requires a Signalling Gateway (or other similar "translator" function), which exceeds the purposes of audio tool development.

Adding ATM compatibility to CATE is complicated due to the lack of an ATM networking API in the official Java platform. Given that, a full ATM library had to be implemented, based on the Java distribution source code, the Java Native Interface (JNI, written in C), the ATM Forum draft specification for Java on ATM [13] and the architecture presented in Figure 5.7. Essentially, the CATE ATM Library (CAL) consists of two APIs: The Java ATM Package (JAP), written in Java, which includes all function calls (e.g., socket creation and QoS specification) for ATM networking; and the Native ATM Layer (NAL), which isolates JAP from the OS-specific ATM networking API. All JAP methods that interact with the network are actually implemented by calls to NAL, which are, in turn, implemented as calls to the (OS-specific) ATM API; this is the model followed by Sun for the implementation of the TCP/IP Java networking API (i.e., the java.net package), although, for simplicity, most of the low-level JVM security checks were omitted.

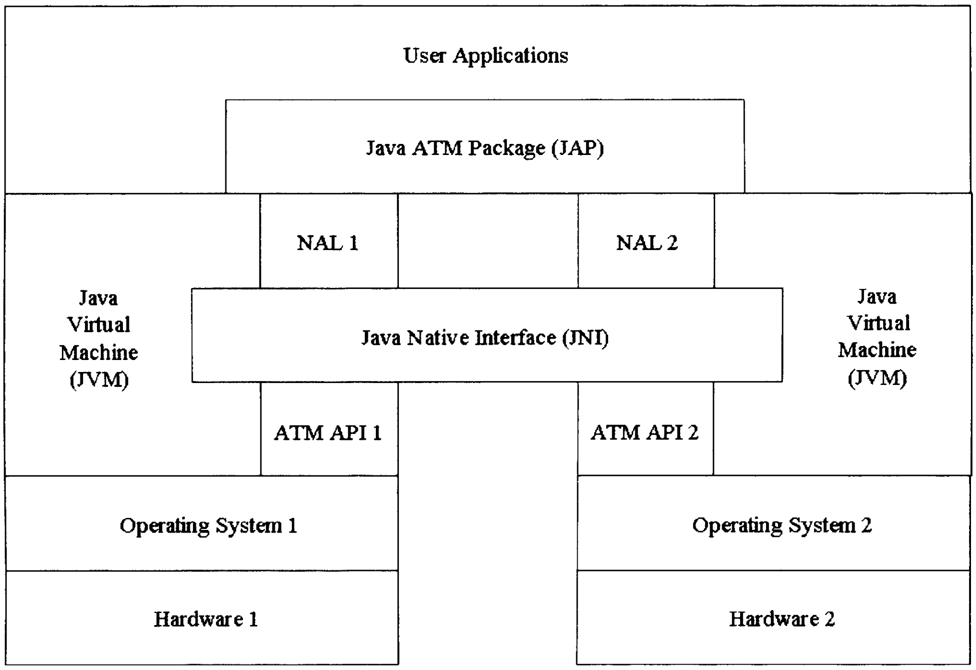


Figure 5.7: JAP over NAL and the JVM, in two different architectures, 1 and 2

JMFE was ATM-enabled by writing a special DataSource class, which essentially is a wrapper for a CAL ATM socket, following the JMF API conventions. Whenever a call is connected, a DataSource object is instantiated, reading and writing AAL1 voice cells until closed by the application.

In RATE, ATM voice capability was added by the development of a (pseudo) audio device, which substitutes the actual (hardware) audio device driver. Like with the JMFE, when a call is established, this device is activated, reading and writing AAL1 cells containing voice samples. As of Version 4.1.3, this functionality has also been incorporated in the official RAT distribution, using the CAT code [U43].

5.4.8 Cross-platform Operation

Taking advantage of the cross-platform nature of Java, as well as the existence of JMF and RAT versions for Windows and Linux, the Configurable Audio Tool was also ported and tested over two major hardware and OS platforms, Windows and Linux (Table 5.3). This excludes the ATM interface, which was fully tested only on the Linux version of CATE, JMFE and RATE, over Kernel 2.4.2 and the ATM-Linux library (Version 0.79).

Technical Characteristic	**Windows Testbed**	**Linux Testbed**
Java Virtual Machine	Sun Java 2, JRE 1.2.2	Sun Java 2, JRE 1.2.2
Operating System	MS-Windows 2000 SP2	RedHat Linux 7.2, Kernel 2.4.2
Computer System	Dell Optiplex GX200	Dell Optiplex GX200
Architecture	Intel	Intel
Processor	Pentium III / 1.2 GHz	Pentium III / 1.2 GHz
Memory	1 GB RAM, 40 GB HDD	1 GB RAM, 40 GB HDD
IP Connectivity	Ethernet, 10/100 Mbps NIC	Ethernet, 10/100 Mbps NIC
ATM Connectivity	-	ATM-on-Linux 0.79, UNI 3.1, 155 Mbps NIC

Table 5.3: CAT Test Platforms

5.5 Evaluation

To evaluate the developed IP Telephony User Agent, two parameters of the implementation were considered, following the methodology presented in Chapter 1: implementation complexity and operational performance. (An additional assessment of the CAT media engine was conducted as part of the VIA evaluation, which can be found in Chapter 6.)

In the first case, the only variable part of the audio tool is the media engine, which, as already discussed, is considered to be the most complicated one. Therefore, the size of customised code needed to add this component into CAT, shown in Table 5.4 for each of the 3 different alternatives, is a good indication of complexity. (CATE and RATE were developed from scratch, whereas for JMFE the code needed to implement a media engine using the existing libraries of the JMF toolkit is quoted.)

For the performance assessment of CAT, measurements of processing delays associated with the most intensive part of the application, the encoding/decoding procedure, were conducted. More specifically, the time spent for encoding (during transmission) and decoding (during reception) a 30 second, pre-recorded, native English, male speech fragment, in ADPCM and LPC, was measured. The VoIP session established used 20 ms payload sizes for the RTP packets produced. Results are plotted in Figure 5.8, in 4 diagrams of delay versus packet cycles. Each cycle corresponds to the entire processing period required for a packet, prior to the next packet being able to be processed, so the values shown must be added to the corresponding ADPCM and LPC CODEC delay (0.125 ms and 22.50 ms, respectively) in order to calculate the total delay experienced by a packet inside CAT, before entering the output buffer of the network interface. The testbed used to run the application in Windows and Linux was an Intel-based mid-end PC, with large memory and a previous generation processor, as detailed in Table 5.3.

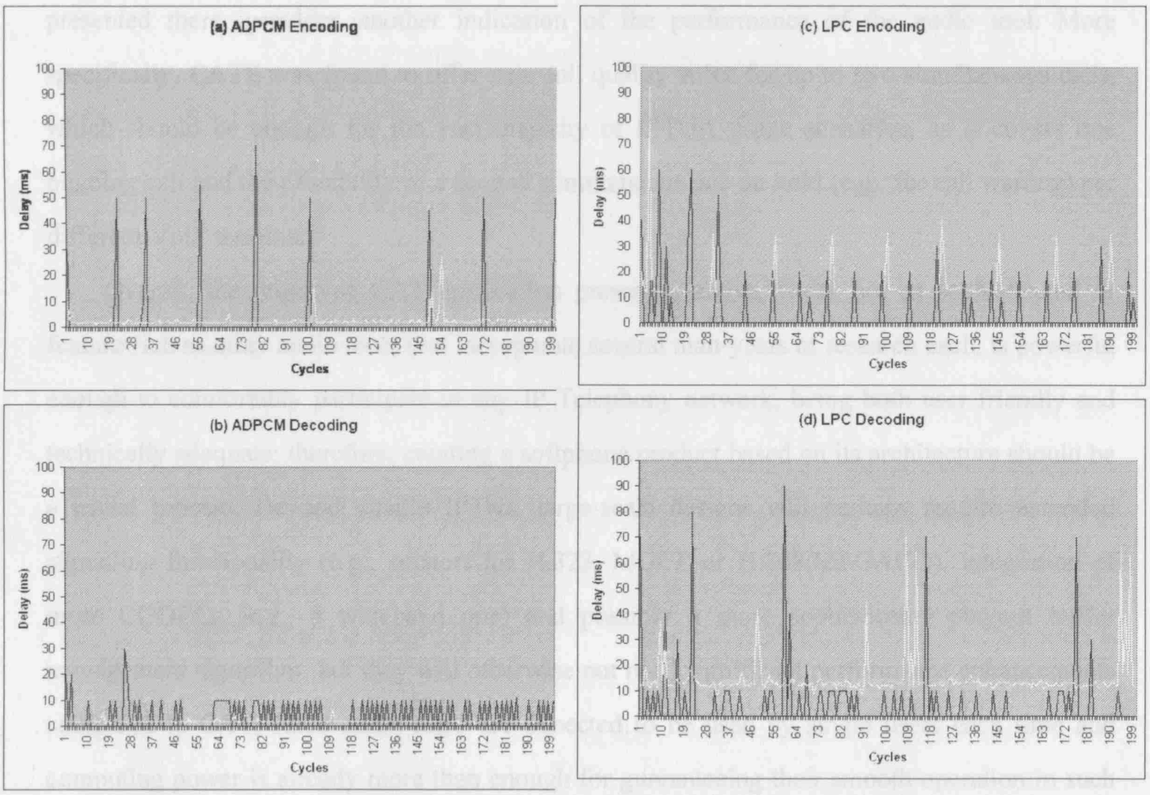


Figure 5.8: CAT performance under ADPCM and LPC

In the case of ADPCM, the Windows version (black line in the diagram) presents periodic spikes, centred around multiples of 10 or 20 cycles for encoding and much more often, leading to trains of pulses, for decoding; by contrast, the Linux version (white line) is characterised by nearly constant delay in both cases. For LPC, the picture is similar with ADPCM under Windows (although spikes become more often), while in Linux the average delay is greatly increased and periodic spikes appear; these happen roughly every 20 cycles and coincide with the corresponding ones of the Windows version, hinting at some sort of correlation. More generally, the performance of the application is constantly better (very small delay, most of the time less than 1 ms) in Windows, compared to Linux (which, on average needs around 2.0 ms to 2.5 ms for ADPCM and 14 ms-14.5 ms for LPC), as shown in Table 5.5.

Although the implementation details of CAT, the Java Virtual Machine and the two Operating Systems involved complicate the analysis of the results, it appears that the spikes in all cases, as well as the large processing delays associated with Linux version of LPC, can be mostly attributed to the JVM implementation. Spikes, in particular, may be caused by a combination of OS scheduling and JVM management (e.g., garbage collection) operations.

The CAT media engine, evaluated in Chapter 6 for use in the LIFT-compliant gateway presented there, provides another indication of the performance of the audio tool. More specifically, CATE was found to offer near toll quality voice for up to two simultaneous calls, which should be enough for the vast majority of IPTUA usage scenarios, as it covers one ongoing call and the possibility of a second simultaneous one on hold (e.g., for call waiting) per different VoIP terminal.

Overall, the prototype CAT application presented above, while not as sophisticated or feature-rich as other audio tools that incorporate several man-years of research each, is powerful enough to comfortably participate in any IP Telephony network, being both user friendly and technically adequate; therefore, creating a softphone product based on its architecture should be a trivial process. Beyond simple IPTNs, large-scale designs will perhaps require extended signalling functionality (e.g., support for H.323, MGCP or H.248/MEGACO), integration of more CODECs (e.g., a wideband one) and possibly a more sophisticated playout buffer management algorithm, but they will otherwise not need significant performance enhancements compared to CAT, since audio tools are expected to be used by single users each time and computing power is already more than enough for guaranteeing their smooth operation in such cases. This, among other things, renders the issue of the hardware employed irrelevant, on the grounds that, since CAT operates comfortably, as already seen, in the demonstration testbed used, it should face no performance issues with contemporary, cutting-edge computer or other similar equipment (e.g., IP Phones).

Media Engine	Lines of code
CATE	3,500
JMFE	350
RATE	30,000

Table 5.4: Media Engine complexity

Experiment	Windows		Linux	
	AVG	STDDEV	AVG	STDDEV
ADPCM Encoding	2.30	10.61	2.51	2.32
ADPCM Decoding	3.76	5.45	2.02	0.56
LPC Encoding	3.50	9.91	13.71	7.26
LPC Decoding	4.80	13.11	14.47	8.67

Table 5.5: Average and standard deviation of delay values

5.6 Conclusion

The current, third generation of audio tools consists mainly of either *softphones* that have sophisticated user interfaces, support a single signalling protocol (typically H.323 or SIP, if not proprietary, like Skinny or Skype) and offer varying levels of media functionality (e.g., certain CODECs, silence suppression and error concealment schemes), or of *audio processes* integrated in conferencing applications (e.g., NetMeeting). At the same time, writing a basic audio tool has become trivial, due to the release of standardised libraries (e.g., JMF) over the past few years.

This chapter has presented a novel IP Telephony User Agent, CAT, which introduces a modular architecture with clearly defined APIs that allow the integration of different signalling protocols and multiple (pluggable) media engines, while achieving performance levels comparable to more sophisticated applications, like RAT. In essence, CAT implements the techniques discussed in Chapter 2 and the LIFT guidelines analysed in Chapter 4, thus being suitable for a number of roles: a software core that can be built in VoIP hardware (e.g., a low-end IP Phone), a stand-alone IP Telephony User Agent, or even a multi-layered application which can be installed in devices of widely varying capabilities, with the ones suitable for each particular device enabled (configured in) at the production or usage phase. Furthermore, the generic audio tool architecture presented herein can be used for the development of other voice applications and, indeed, the pluggable media engine concept is further exploited in Chapter 6, for the implementation of the media part of the corresponding gateway.

CHAPTER 6

An IP Telephony Gateway

6.1 Overview

The implementation of a global, highly integrated voice network, replicating the current GSTN feature set and offering new services (e.g., instant messaging or conferencing) to end users, mandates interoperability among different communication technologies, including (but not restricted to) circuit and packet switching. Special interworking devices (gateways) constitute a popular mechanism for achieving this goal and, as VoIP matures, they are increasingly diversifying their functionality to encompass most signalling and media components of an IPTN, as described earlier on.

This chapter presents a novel gateway architecture and the design, implementation and evaluation of a prototype device that follows it, while also complying with the functional requirements set by LIFT in Chapter 4 and exploiting a significant amount of earlier experience and research output derived from the 2-year Voice over IP and ATM gateway (VIA) project of UCL-CS and Nortel. The performance of the implemented gateway is found similar to that of a conventional (TIPHON) one, under the same software and hardware environment, thus further supporting the viability of LIFT. This way, the second of the three representative areas identified in Chapter 1 is investigated and the corresponding part of the hypothesis is accordingly supported.

6.2 Related Work

Gateways are traditionally seen as a transparent solution to the problem of bridging heterogeneous communication systems [35], [289], [300]. Thus, their use for IP Telephony, in particular, has been extensively researched since the 1990s, when the proliferation of VoIP brought forward an increasing need for interoperation with SCNs like the GSTN [146], [159], [394]. The first generation of VoIP gateways roughly coincided with the second generation of audio tools discussed in Chapter 3 and offered support for up to a few hundreds or thousands of media ports per device; however, due to the transitional status of the main IP Telephony protocols (H.323, SIP, RTP) and the relative immaturity of the technology, interoperability with the telephony network was limited to the basic features and scalability was hampered by the integrated (all-in-one), monolithic designs of the time [6], [80], [85], [164], [277], [295]. Ongoing research efforts produced the first decomposed gateway architectures that appeared in the late 1990s [166] and were later finalised into the current, second generation, distributed architecture introduced by the TIPHON model [107], analysed in Chapters 3 and 4.

Several recent publications describe the design and implementation of decomposed VoIP gateways offering bidirectional translation from GSTN to IP Telephony at the backbone [61], [85], [89], [399] and virtually all major commercial vendors include a scalable such gateway in their product portfolio [79], [159], [237], [U23]. While academic work focuses on the architectural aspect and usually addresses the entire decomposed gateway, commercial devices are essentially Media Gateways offering up to a few hundreds or thousands of ports each that, combined with softswitch products from the same or different manufacturers, can scale up to one or two orders of magnitude more [281], [U7], [U27], [U30], [U33], [U59]. In both cases, the gateways described perform a common set of tasks, to various extents of implementation detail, feature richness and performance: they convert between common signalling (Q.931 or SS7, to H.323, SIP or both) and media (TDM to RTP) formats in both directions; they use standardised mechanisms (SCTP and MGCP or H.248/MEGACO) for decomposition; they support multiple voice CODECs; and they employ dedicated hardware (e.g., custom DSPs and fast processing modules) to speed up operations for scalability purposes. In a smaller scale, various types of residential gateways (e.g., home routers, xDSL or cable modems, wireless access points, set-top boxes and game consoles) exist for delivering network services over the last mile, and a convergence process towards integrated multimedia communication is ongoing, leading to products that range in complexity and capabilities from small VoIP boxes to devices that offer simultaneous access to multiple different networks, links (wired or wireless) and media [18], [47], [92], [382].

A common characteristic of second generation gateways remains their isolated operational model, due to the lack of support for a capability announcement mechanism, since the corresponding IETF protocols (TGREP [17]/TRIP [314], or a similarly suitable method such as SLP [141]) have not (at least yet) been widely adopted. Overall, however, it can be argued that the concept of the IP Telephony gateway has already matured, leading to a broad consensus on what such a device is and what it should do inside a network. As seen earlier on, the framework developed in this thesis, LIFT, exploits this maturity and proposes additional improvements to the gateway concept, for scalability purposes; these improvements are incorporated in the Voice over IP and ATM Gateway (VIA), analysed in this chapter.

The particular differences between conventional (TIPHON-based) and LIFT-compliant gateways are presented in Chapter 4. More specific details about the technologies discussed in this section and the rest of the chapter can be found in the corresponding parts of the literature review, presented in Chapters 2 and 3.

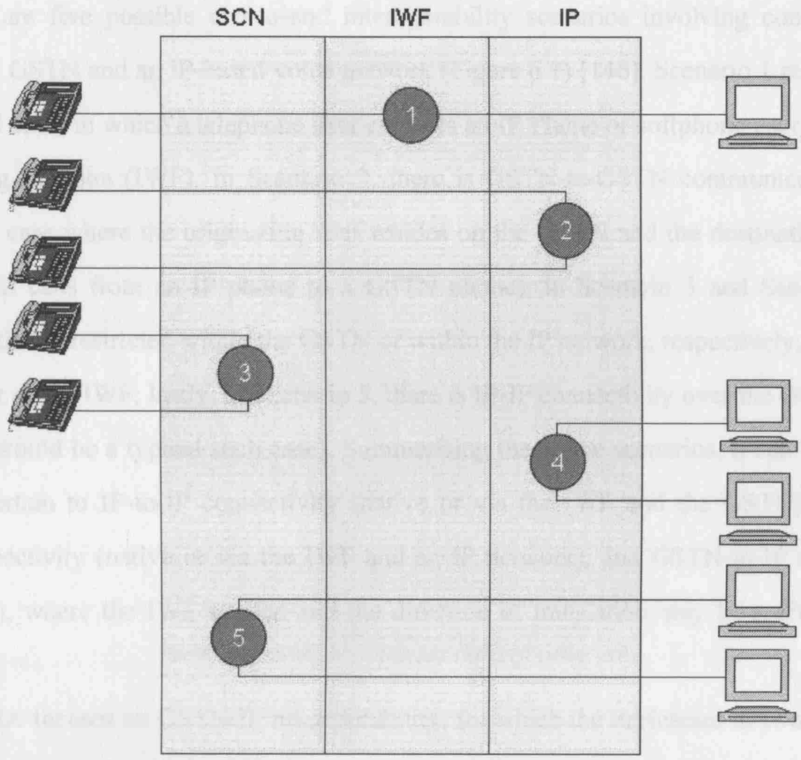


Figure 6.1: SCN-IP Interoperability Scenarios

6.3 Architecture and Design

The design of large-scale, heterogeneous, interoperable IP Telephony networks (IPTNs) must take into account the two main formats for transporting voice traffic (i.e., speech and other audio types, such as voice-band tones):

(a) *Raw* (native): Analogue or digital voice, carried over a Switched Circuit Network (SCN), which is usually the General Switched Telephone Network (GSTN), or a mobile telephony network (e.g., GSM-based).

(b) *Packet*: Digital voice, carried over a packet switched network, most commonly based on IP, Frame Relay or ATM, with VoIP being the most popular example.

In two characteristic cases of SCN and packet networks, the GSTN and VoIP, the digital format has prevailed. More specifically, as seen in earlier chapters, analogue voice transmission is being abandoned across the GSTN [26], while for digital voice the packet format is becoming increasingly attractive, particularly over IP [128]; in fact, the raw format is being pushed at the edges of the network, since packet technologies like IP and ATM are already used for voice transportation in the GSTN backbone [394].

There are five possible end-to-end interoperability scenarios involving communication between the GSTN and an IP-based voice network (Figure 6.1) [146]. Scenario 1 represents the most typical case, in which a telephone user contacts an IP Phone or softphone user, through an interworking function (IWF). In Scenario 2, there is GSTN-to-GSTN communication via IP (e.g., in the case where the originating user resides on the GSTN and the destination user has redirected all calls from an IP phone to a GSTN phone); in Scenario 3 and Scenario 4, the communication is restricted within the GSTN or within the IP network, respectively, without the involvement of the IWF; lastly, in Scenario 5, there is IP-IP connectivity over the GSTN (again, redirection would be a typical such case). Summarising these five scenarios, it can be observed that they pertain to IP-to-IP connectivity (native or via the IWF and the GSTN), GSTN-to-GSTN connectivity (native or via the IWF and an IP network), and GSTN-to-IP connectivity (Scenario 1), where the IWF is used and the direction of translation may be a differentiating factor [316].

The VIA focuses on GSTN-IP interoperability, for which the intricacies of two major (and significantly different) network technologies need to be considered:

(a) The GSTN, a type of SCN with [26]

- Analogue PSTN (also called the Plain Old Telephone Service, POTS)
- Digital PSTN (replacing POTS)
- Narrowband ISDN (N-ISDN)
- Broadband ISDN (B-ISDN), based on ATM

(b) IP networks, based on packet switching, with [128]

- Best effort operation
- Enhanced operation (through traffic engineering)
- QoS operation (through Integrated Services, Differentiated Services or MPLS)

As already discussed, transparent communication between the GSTN and IP networks mandates the deployment of special interworking functions (IWFs), capable of translating information flows at heterogeneous network borders. The top-level design of such a device (essentially a LIFT-compliant distributed gateway) and its role within a combined GSTN/IP voice network is pictured in Figure 6.2, while details of the network itself and of the major components involved, are provided in the following sections.

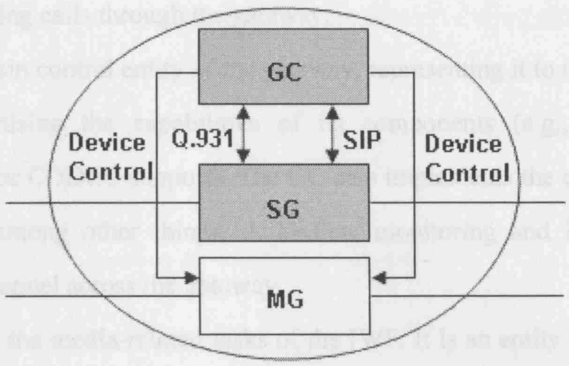


Figure 6.2: A generic GSTN-IP interworking function (IWF)

6.3.1 Network Model

The proposed gateway architecture interconnects the GSTN with a Voice over IP network, by translating among the communication protocols and formats involved. As shown in Figure 6.2, this is a combined task performed by three major components, the Signalling Gateway (SG), the Gateway Controller (GC) and the Media Gateway (MG), in compliance with the main VoIP frameworks, where the GC is referred to as the MGC (Media Gateway Controller). The same architecture is also compliant with LIFT, as seen in Chapter 4, where the GC is used as a more generic entity.

On the signalling path, the GSTN protocol is assumed to be Q.931 [201] and the IP one either H.323 [188] or SIP [318]; The traffic path carries Constant Bit Rate (CBR) voice on the GSTN side (typically in TDM format, e.g. T1 or E1) [26] and Variable Bit Rate (VBR) voice (encapsulated in RTP packets over UDP [286]) on the IP side. Inside the gateway, communication among the components is implemented over IP links, and the protocols involved depend on the implementation, as analysed in the next section.

6.3.2 Main Components of the Gateway

The three main components of the gateway are the Signalling Gateway (SG), the Gateway Controller (GC) and the Media Gateway (MG), as shown in Figure 6.3.

The SG is responsible for the signalling part of the IWF. It terminates GSTN signalling (Q.931), encapsulates necessary information and passes it over to the GC (acting in this case as a MGC) over the IETF standard for signalling transport (SIGTRAN, solid SG-GC line in Figure 6.3); alternatively, the SG can perform the translation locally and inform the GC using native

VoIP signalling (dashed SG-GC lines in Figure 6.3). In both cases, the GC is ultimately responsible for admitting calls through the gateway.

The GC is the main control entity of the gateway, representing it to the external network by collecting and advertising the capabilities of its components (e.g., signalling protocols, connection resources or CODEC support). The GC also implements the call control logic of the gateway, including, among other things, admission, monitoring and instructing the MG to establish the media channel across the gateway.

The MG handles the media-related tasks of the IWF. It is an entity that translates between GSTN voice (typically in TDM format, e.g. T1 or E1) and IP voice (encapsulated in RTP packets over UDP); the translation process may include transcoding, given the narrow selection of CODECs offered by the GSTN, where usually only G.711 is used (and G.726 for international calls) [26]. The MG operates as a slave function to the GC, which controls it by using an IP connection and a special protocol for this purpose (either a Proprietary Device Control protocol, or a standardised one, such as MBUS [284], MGCP [5] and H.248/MEGACO [136], [187]), similarly to the architectures presented in Chapters 3 and 4.

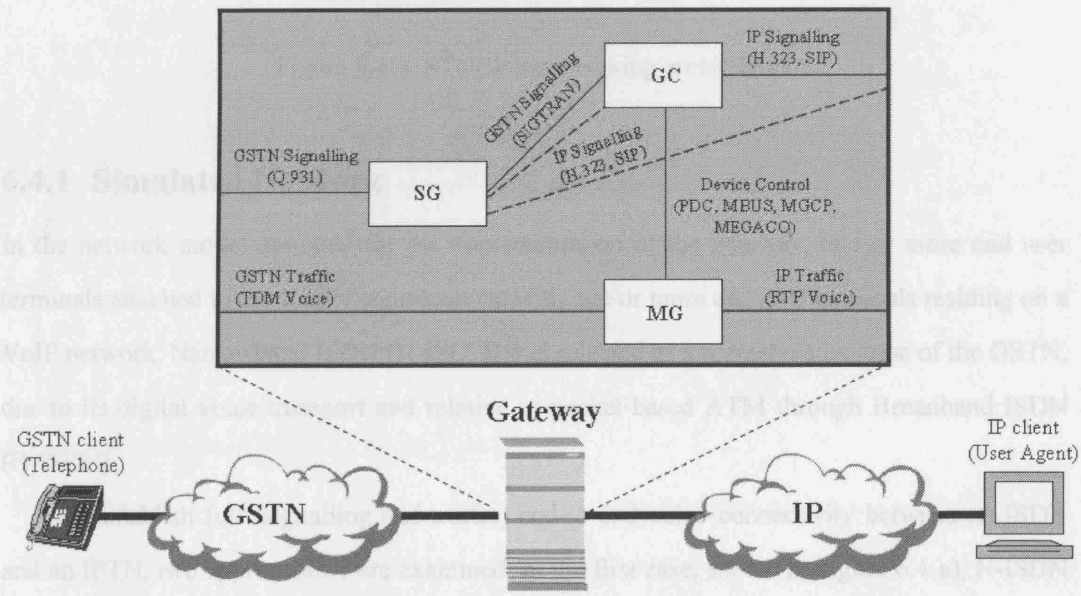


Figure 6.3: Main components of a VoIP gateway

6.4 A Research Implementation

An instance of the architecture analysed in the previous section has been implemented, as a proof of concept, into a simple VoIP Gateway. The device developed is called the Voice over IP and ATM Gateway (VIA), because of the network model used during simulations, as explained in the next sections.

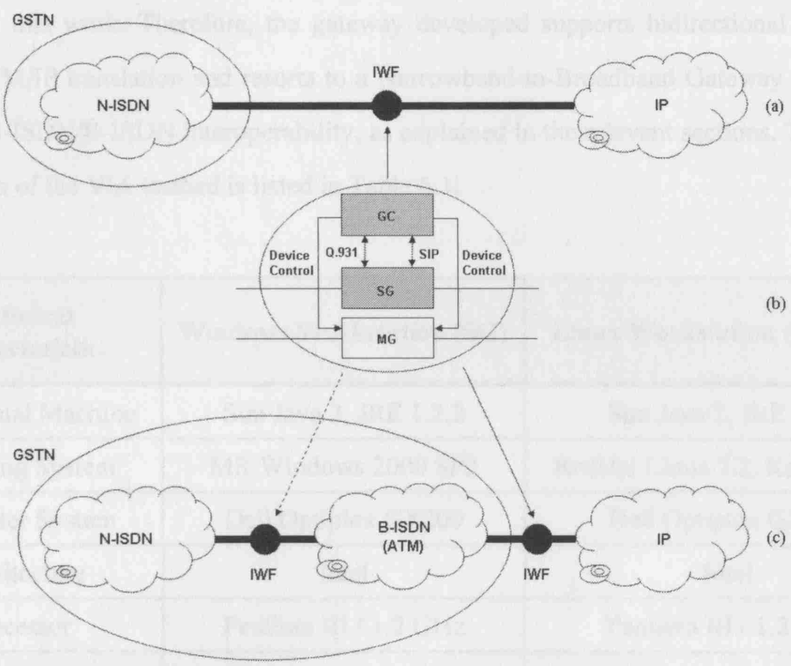


Figure 6.4: GSTN-IP interworking architecture

6.4.1 Simulated Network

In the network model assumed for the implementation of the gateway, one or more end user terminals attached to the GSTN communicate with one or more end user terminals residing on a VoIP network. Narrowband ISDN (N-ISDN) was selected as a representative case of the GSTN, due to its digital voice transport and relation to packet-based ATM through Broadband ISDN (B-ISDN).

To establish full (signalling and traffic) end-to-end voice connectivity between an ISDN and an IPTN, two approaches were examined. In the first case, shown in Figure 6.4(a), N-ISDN connects to IP and the only intermediate mechanism between the two networks is the IWF; here, in addition to signalling and voice protocol translation, special mechanisms (e.g., dedicated hardware) would be needed for adapting (packetising/depacketising) the voice stream. For the second case, shown in Figure 6.4(c), N-ISDN traffic is relayed to B-ISDN (ATM), which then connects to IP, meaning that the adaptation of the voice stream is handled by the ATM equipment and so the corresponding extra work is not needed; however, another IWF must be used for linking N-ISDN with B-ISDN.

The second approach is more interesting, because, in addition to narrowband voice support, it also achieves compatibility with Voice over ATM (VoATM) networks, and so it was followed in this work. Therefore, the gateway developed supports bidirectional GSTN-to-IP calls by ATM/IP translation and resorts to a Narrowband-to-Broadband Gateway (NG) for the necessary N-ISDN/B-ISDN interoperability, as explained in the relevant sections. The technical specification of the VIA testbed is listed in Table 6.1.

Technical Characteristic	**Windows Workstation (SG)**	**Linux Workstation (GC, MG)**
Java Virtual Machine	Sun Java 2, JRE 1.2.2	Sun Java 2, JRE 1.2.2
Operating System	MS-Windows 2000 SP2	RedHat Linux 7.2, Kernel 2.4.2
Computer System	Dell Optiplex GX200	Dell Optiplex GX200
Architecture	Intel	Intel
Processor	Pentium III / 1.2 GHz	Pentium III / 1.2 GHz
Memory	1 GB RAM, 40 GB HDD	1 GB RAM, 40 GB HDD
IP Connectivity	Ethernet, 10/100 Mbps NIC	Ethernet, 10/100 Mbps NIC
ATM Connectivity	-	ATM-on-Linux 0.79, UNI 3.1, 155 Mbps NIC
GSTN PBX (NGS)	Nortel Meridian 1	
ATM Switch (NGM)	Marconi ASX 200	

Table 6.1: The VIA testbed

6.4.2 Signalling and Traffic Profiles

On the control plane, the VIA acts as an IWF between GSTN and IP signalling flows by translating from Q.931 to SIP, and vice versa; SIP signalling was selected, instead of H.323, due to its relative simplicity, ease of implementation, wide acceptance within and outside the IETF, native bias towards IP, functionality, and for a variety of other reasons, analysed in Chapter 3; this approach is also favoured by the rough correspondence of the basic Q.931 and SIP message types (Table 6.2) and has been exploited in earlier work [367]; for internal signalling, a proprietary event-based mechanism is used. On the user plane, voice is encoded in the typical N-ISDN format (64 Kbps PCM A/u-law G.711) and transported as CBR traffic over ATM

AAL1 [190] VCCs for the GSTN part; on the IP side, a number of CODECs are supported, producing CBR traffic to be sent over VBR RTP/UDP/IP links in the IP part.

Q.931 Messages	SIP Messages	PDC Messages
SETUP	INVITE	START DATA TRANSFER
CALL PROCEEDING	180 Trying	START TIMER
ALERTING	181 Ringing	STOP TIMER
CONNECT	200 OK	STATUS
CONNECT ACK	ACK	
DISCONNECT	BYE	
RELEASE	200 OK	
RELEASE COMPLETE	-	RELEASE

Table 6.2: Event types (messages) used in the VIA

Figure 6.5 presents the VIA interconnecting the GSTN with an IPTN, along with the main signalling and traffic protocols involved in the end-to-end communication.

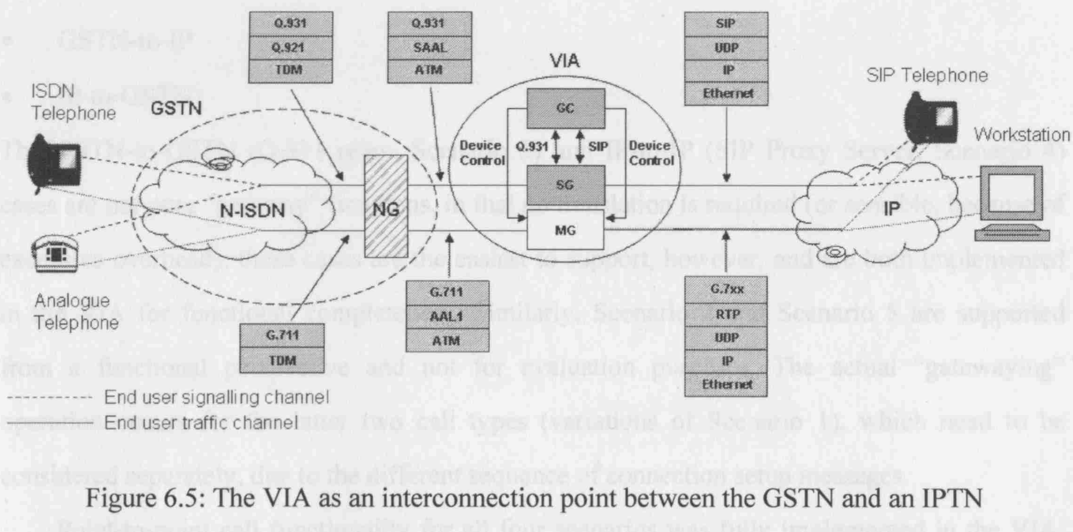


Figure 6.5: The VIA as an interconnection point between the GSTN and an IPTN

6.4.3 VIA Versions

The VoIP gateway designed follows a decomposed (distributed) architecture, which, as seen in Chapters 3 and 4, can be realised in two ways, based on the TIPHON or LIFT frameworks. Accordingly, two versions of the VIA were actually developed, VIA-T (for TIPHON) and VIA-

L (for LIFT), differing, from a functional perspective, in the role of the Signalling Gateway and, consequently, the Gateway Controller components. More specifically, these two versions were constructed as

VIA-T = {SG-T, GC-T, MG}

VIA-L = {SG-L, GC-L, MG}

Enabling or disabling modular Q.931 Agent, Signalling Translation Agent and SIP User Agent objects implemented in both the SG and the GC, along with automatically applied configuration profiles, simplifies switching between the two versions of the VIA, without affecting the operation of the MG. Communication among the gateway components happens over IP links, using a clearly defined, object-oriented Application Programming Interface (API), which integrates the necessary protocol primitives and implements the procedures defined in the corresponding protocol specifications.

The VIA call control kernel is located at the GC and provides full support for single or multiple calls between SCN terminals (e.g., phones) and IP terminals (e.g., workstations). According to the signalling protocols employed at the respective endpoints, there are four possible types of such calls:

- GSTN-to-GSTN
- IP-to-IP
- GSTN-to-IP
- IP-to-GSTN

The GSTN-to-GSTN (Q.931 relay, Scenario 3) and IP-to-IP (SIP Proxy Server, Scenario 4) cases are not pure "gateway" functions, in that no translation is required (or sensible, because of excessive overhead); these cases are the easiest to support, however, and are both implemented in the VIA for functional completeness. Similarly, Scenario 2 and Scenario 5 are supported from a functional perspective and not for evaluation purposes. The actual "gatewaying" operation occurs for the latter two call types (variations of Scenario 1), which need to be considered separately, due to the different sequence of connection setup messages.

Point-to-point call functionality for all four scenarios was fully implemented in the VIA. Point-to-multipoint and multipoint-to-multipoint calls are also supported through the establishment of multiple point-to-point calls and, in addition, UDP multicasting on the IP side.

6.4.4 VIA Components

The main components implementing the VIA are described in the following sections.

6.4.4.1 Signalling Gateway (SG)

The SG of the VIA is implemented in two versions, TIPHON (SG-T) and LIFT (SG-L), both of which exchange signalling information (Q.931) with the B-ISDN/ATM side of the GSTN network (i.e., with the NG). The SG-T forwards the higher layer information (the Q.931 messages) to the GC via a SIGTRAN-based IP connection, whereas the SG-L translates from Q.931 to SIP and informs the GC using SIP. Call processing for both versions of the SG is done on a "per-connection" (point-to-point call) basis. Each point-to-point call is modelled by a special Call Processing Object (CPO), which consumes incoming signalling messages (Q.931 or SIP), schedules them in a single FIFO queue, and processes them sequentially by a dedicated Event Handler thread.

The SG is implemented as a C application running over a Microsoft Windows 2000 PC (Table 6.1).

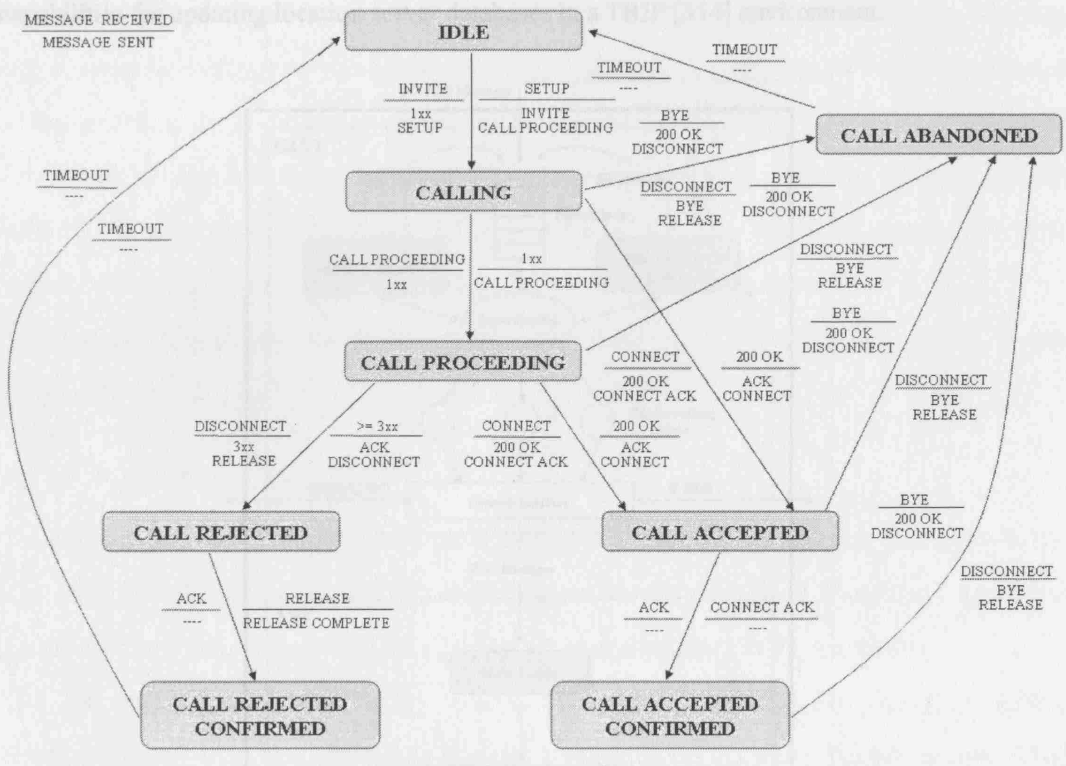


Figure 6.6: The VIA state machine

6.4.4.2 Gateway Controller (GC)

The GC of the VIA is implemented in two versions, TIPHON (GC-T) and LIFT (GC-L), which co-operate with SG-T and SG-L, respectively, as already discussed. The GC-T, in particular, acts as both a Q.931 Agent and a SIP User Agent, according to the state machine produced by the combination of the corresponding protocol automata, as shown in Figure 6.6; the GC-L, on the other hand, operates as a SIP User Agent, capable of initiating and accepting calls on the IP side of the network, since the signalling translation is carried by the SG-L. The UDP version of SIP was implemented for either case, with the additional reliability mechanisms (including retransmission timers) mandated by the specification of the protocol [318].

Both GC versions use the same master-slave mechanism to control the traffic channels that traverse the gateway, through the MG. More specifically, the GC manages the resources (connection pool) of the MG by exchanging Proprietary Device Control (PDC) messages, an approach which was preferred as simpler and easier to implement than using a full-blown protocol for this purpose (such as MBUS, MGCP or H.248/MEGACO). PDC message types (Table 6.2) include authorisation for data transfer, timer actions, analytic status information (such as availability of connections) and the option to release one or more calls (for example, as a reaction to an administration decision). A simple TGREP [17] client gathers gateway capabilities for updating location server databases in a TRIP [314] environment.

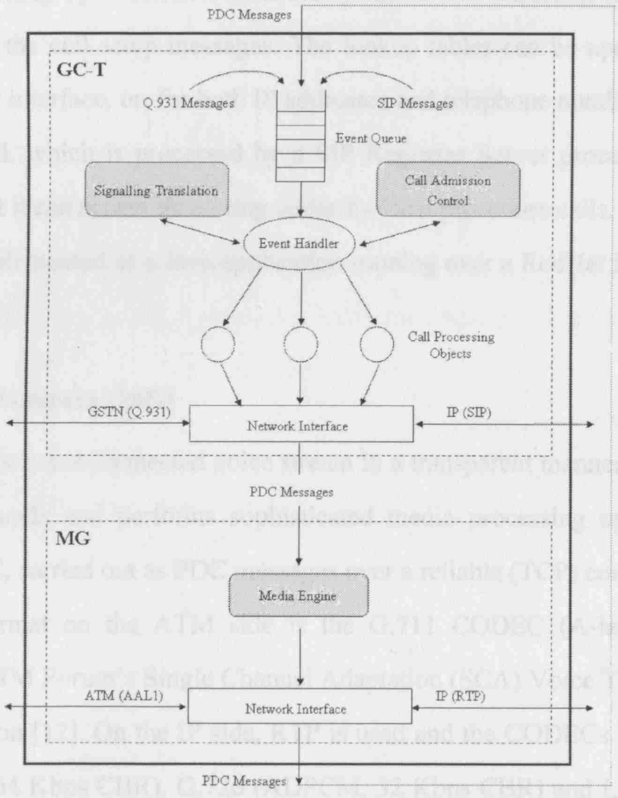


Figure 6.7: Event handling inside GC-T

Call processing is done at the GC by modelling each point-to-point call as a special Call Processing Object (CPO), which consumes the events (Q.931, SIP or PDC) that are addressed to it. All the events are scheduled in a single FIFO queue, and processed sequentially by the GC Event Handler object, which owns it (Figure 6.7). For each event received by the call object, the actions taken include a possible state transition, as well as the creation of one or more messages, as a response to the sending endpoint and/or a notification to the receiving endpoint. For example, on a GSTN-to-IP call, the incoming Q.931 SETUP message is translated into a SIP INVITE, which is transmitted to the called workstation; at the same time, a Q.931 CALL PROCEEDING is returned to the caller (Figure 6.6). Events directly related to new call requests are assessed by a Call Admission Control (CAC) entity in the GC, which, for the purposes of this implementation, is restricted to source/destination address checks and resource availability verification. Every accepted call is handled by a corresponding CPO, which implements its own finite state machine (Figure 6.7).

The GC implements the address translation function of the VIA; addressing is based on SIP, which defines a URL syntax for both GSTN and IP clients [375]. The GC also performs call routing, by looking up a number of directory tables and matching the source/destination addresses found in the call setup messages. The lookup tables can be updated either by local entry, from the user interface, or, for both IP addresses and telephone numbers, by using the SIP REGISTER method, which is processed by a SIP Registrar Server process that is collocated with the GC, so that it can access its routing tables by local procedure calls.

The GC is implemented as a Java application running over a RedHat Linux 7.2 PC, as can be seen in Table 6.1.

6.4.4.3 Media Gateway (MG)

The MG manages the cross-connected voice stream in a transparent manner. It terminates ATM and IP traffic channels and performs sophisticated media processing operations, under the directions of the GC, carried out as PDC messages over a reliable (TCP) connection.

The traffic format on the ATM side is the G.711 CODEC (A-law, 64 Kbps CBR), complying to the ATM Forum's Single Channel Adaptation (SCA) Voice Telephony over ATM (VTOA) specification [12]. On the IP side, RTP is used and the CODECs supported are G.711 (A-law and u-law, 64 Kbps CBR), G.726 (ADPCM, 32 Kbps CBR) and LPC (5.6 Kbps CBR) [229]. The MG was written in pure Java, which, among other things, mandated the development of a full Java ATM networking library, as the language currently only supports TCP/IP. For this purpose, the OOP model of Sun's Java 2 distribution (SDK 1.2) was followed, the C-based Java Network Interface (JNI) was used and a draft specification of the ATM Forum for ATM networking in Java [13] was implemented.

The MG application software has two main components, a control and a data one. The control component is responsible for call processing, according to the interaction of the MG with the GC, as well as for monitoring the data exchange over the network interfaces of the MG. The lightweight, low-level signalling necessary for the establishment of the actual traffic connections (abstracted at a programmatic level by ATM and UDP sockets) is also part of the duties of this component. The data component acts as the core Media Engine, which performs all the actual media processing in the MG. Its main tasks are re-packetisation (conversion between ATM cells and IP packets), transcoding (translation of available encoding formats), silence removal, error control and synchronisation. A special data structure, the PlayThrough Buffer, is used to restore timing dependencies between the CBR (ATM) and the VBR (IP) sides of the MG.

The actual reading of AAL1 voice samples from the ATM side of the network is achieved via directly accessing raw ATM cells (AAL0) through Java ATM sockets. The source clock recovery and partial filling options of AAL1 were not used, which resulted in increased bandwidth efficiency (because of the full 47-byte payload being available), at the unavoidable expense of increased packetisation delay (which, however, due to the scale of the experiment, did not create any perceivable problems such as echo). This approach has the additional benefit that no extra Convergence Sublayer Information is needed, hence the CSI bit is always set to 0, forcing the AAL1 PDU header to assume only 8 different values, in a circular ("ring") fashion, as shown in Figure 6.8.

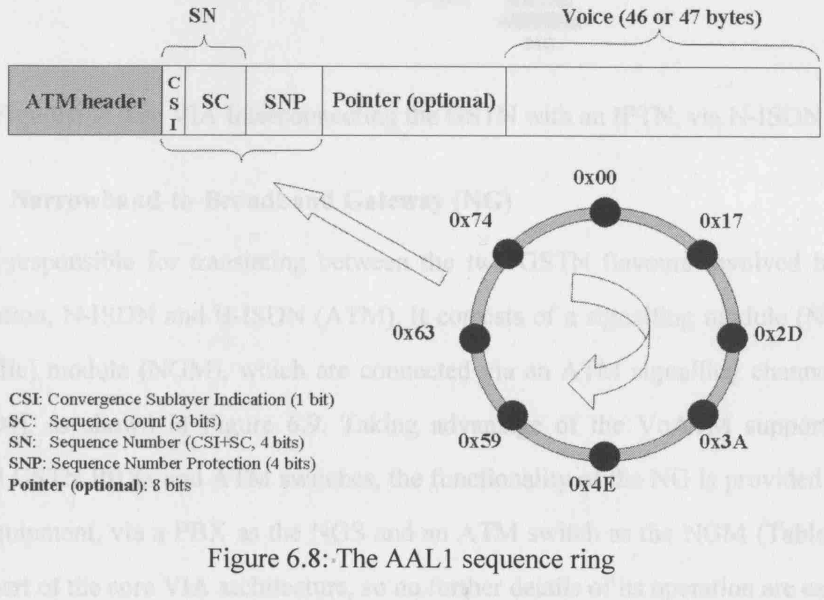


Figure 6.8: The AAL1 sequence ring

The thin interface joining the data component with its control counterpart allows the use of different media engine implementations in the VIA. Similarly to the work presented in Chapter 5, three different such modules have been tested: The CAT Engine (CATE), part of the Configurable Audio Tool analysed earlier on, the JMF Engine (JMFE) from a custom audio tool (Appendix A) based on the Sun Java Media Framework ("JMF") Version 2.1 (FCS) [U51], and the RAT Engine (RATE), essentially Version 3 of the Robust Audio Tool [154], [163], [U43] executed in "stealth" mode (i.e., run without its user interface).

CATE and JMFE are pure Java applications, hence the developed Java ATM Library was used for ATM connectivity in both cases; RATE is based in C, so a custom pseudo-audio device was written, for reading and writing PCM A-law speech samples from/to the ATM network in the form of AAL1 cells, hence effectively replacing the hardware audio interface (sound card) that RATE expects to interact with when used as a stand-alone audio tool. As of Version 4.1.3, this pseudo-audio device has also been incorporated in the official RAT distribution [U43].

The MG is implemented as a Java application (with portions of C code, where necessary), running over a RedHat Linux 7.2 PC (Table 6.1).

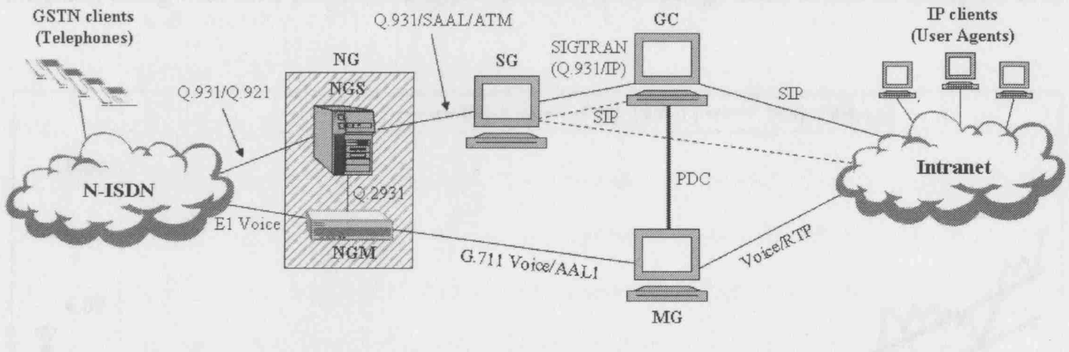


Figure 6.9: The VIA Interconnecting the GSTN with an IPTN, via N-ISDN

6.4.4.4 Narrowband-to-Broadband Gateway (NG)

The NG is responsible for translating between the two GSTN flavours involved in the VIA implementation, N-ISDN and B-ISDN (ATM). It consists of a signalling module (NGS) and a media (traffic) module (NGM), which are connected via an ATM signalling channel, running Q.2931 [204], as shown in Figure 6.9. Taking advantage of the VoATM support found in commercial GSTN PBXs and ATM switches, the functionality of the NG is provided using off-the-shelf equipment, via a PBX as the NGS and an ATM switch as the NGM (Table 6.1). The NG is not part of the core VIA architecture, so no further details of its operation are examined.

6.5 Evaluation

The overall VIA performance has been assessed, comparatively, in a lab environment, through two sets of experiments, one for signalling and one for media, following the methodology presented in Chapter 1. In both cases, GSTN-to-IP and IP-to-GSTN point-to-point calls were examined, over a LAN, without any significant amounts of other traffic being present.

For the first experiment, signalling latencies were measured using a set of 10 different sequences of 100 call setup messages each, distributed over a 1-minute period according to a Poisson model (λ=100) [26], as dictated by a random traffic generator application; in addition, the call originator locations were evenly split (i.e., 50% GSTN-IP and 50% IP-GSTN in every sequence). The resulting calls were injected to both versions of the VIA (VIA-T, VIA-L) and, for each version, the resulting delays (i.e., the times from the reception of the setup message to the opening of the media channel through the gateway) were recorded. No major performance differences were identified among the 10 elements of the set members for each version, so the corresponding average values were calculated and are plotted for both versions in a single diagram, along with their polynomial approximation (smoothing) lines, as shown in Figure 6.10.

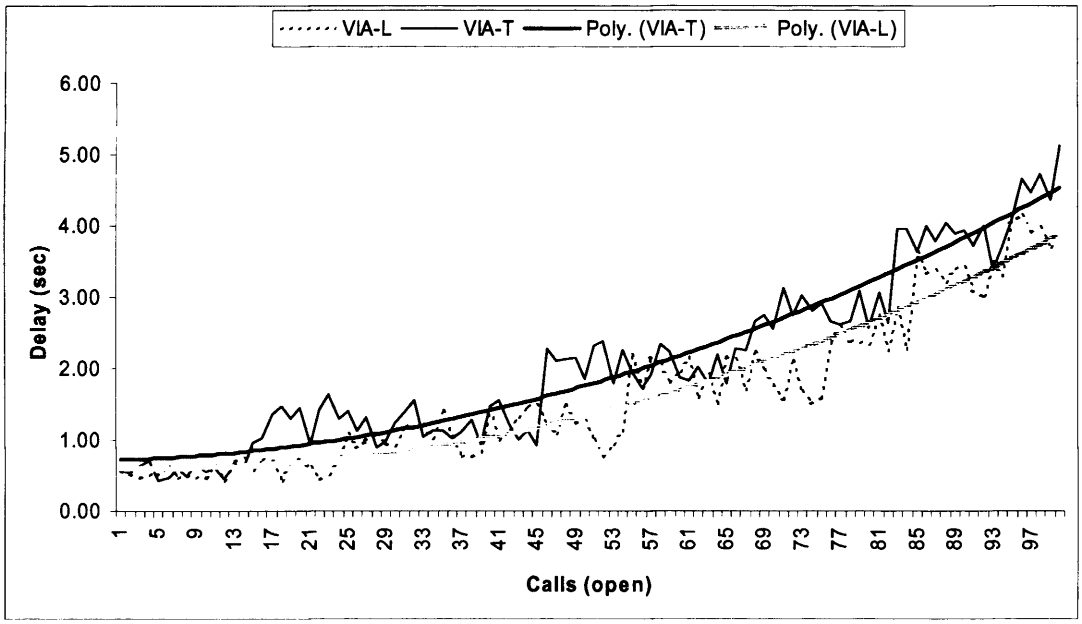


Figure 6.10: Performance of the VIA (VIA-T vs. VIA-L)

Based on the above, the applicability of the LIFT-compliant gateway architecture can be assessed by contrasting its implementation (VIA-L) to the TIPHON-compliant one (VIA-T), which was developed in the same technical environment and under identical resource availability conditions. Interestingly, Figure 6.10 reveals no major performance differences between VIA-T and VIA-L, for most of the time; this was rather anticipated, however, given that, essentially, the two applications only differ by the location of the GSTN-IP signalling translation function (placed in the GC for VIA-T and in the SG for VIA-L). This resemblance indicates, ultimately, that the LIFT architecture should indeed be equally viable to its TIPHON counterpart; actually, in the above experiment, VIA-T even appears to cause consistently higher delays during the call setup procedure, thereby strengthening the assumption that the LIFT design is of at least equal potential to the TIPHON one.

On a more technical approach, given the similarities in the implementation and in the delay patterns evident throughout the diagram, the mismatch seen in Figure 6.10 can be partially attributed to the extra work needed by SG-T to process and transmit Q.931 information to the GC-T, which then translates from Q.931 to SIP; such a processing is unnecessary for VIA-L, where the SG-L does the translation locally. Figure 6.10 shows, furthermore, that less than 15% of calls were set up in around 1 sec, a further 30% (40% for VIA-L) in less than 2 sec and the remaining approximately 55% (45% for VIA-L) in rapidly deteriorating times, even approaching 5 sec, which is unacceptable and can lead to connection timeouts on the GSTN side, but can often be encountered in VoIP environments (e.g., Skype [U48] or other softphones). These delays are attributable to a variety of reasons, mainly implementation-dependent (including performance of various software functions in C and Java, code optimisation, OS scheduling and thread synchronisation). At any rate, since the two designs present strong similarities in performance, they will be examined together henceforth.

For the second experiment, media performance was assessed by subjecting VIA calls to a set of Mean Opinion Score (MOS) test procedures. More specifically, 22 subjects with a computing or engineering background, but low or no exposure to VoIP, were asked to participate in calls, in rotating groups of 5, so that up to 5 simultaneous conversations, each encoded in G.711 (A-law) and lasting for 1.5-2 minutes, were in progress each time; the scenario was repeated for each of the 3 Media Engines (CATE, JMFE, RATE) integrated in the VIA and the overall voice quality of each call was ranked in the MOS scale of 0 (worst) to 5 (best). Results were averaged and are collectively shown in Figure 6.11.

As expected, the RATE Media Engine was found slightly better than the JMFE one, and both clearly outperformed CATE, which, however, still offers near-toll quality voice for up to two simultaneous calls. (Such a finding is also very interesting for the evaluation of the CAT IPTUA, because most of the time such an application is engaged in at most two simultaneous calls, one active and another one on hold, as discussed in Chapter 5.) The relative ranking of the three Media Engines presented above is also on a par with the amount of effort placed in developing them, as well as with the amount of built-in functionality they incorporate.

Overall, the prototype implementation presented above and its measured performance characteristics, indicate the applicability of LIFT as presented in Chapter 4. The VIA is not, of course, as powerful as high-end commercial products, which can support a magnitude more of simultaneous calls and can be combined into multi-rack devices with thousands of media ports. However, such products are optimised versions of the same design as the TIPHON version of the VIA, which was developed and evaluated under identical conditions to the LIFT version, so it can be expected that the latter will be similarly scalable as the former; in addition, commercial gateways implement much of their core functionality in dedicated hardware and integrate tens of man-years of research and development. Given the above, a performance upgrade to the VIA up to industrial levels should be feasible, provided that the proper amounts of time and effort are dedicated to such a purpose.

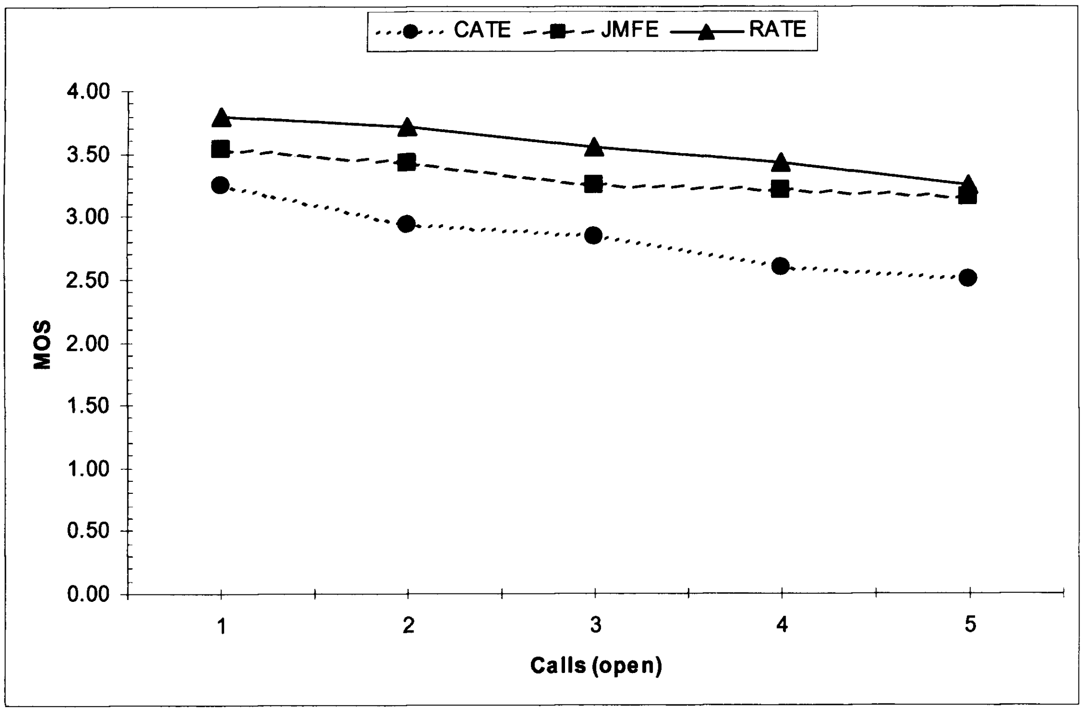


Figure 6.11: Media performance of the VIA under different Media Engines

6.6 Conclusion

Interoperability with the GSTN is essential for the successful deployment of large-scale IP Telephony networks (and, eventually, the ISTN), so the development of efficient interworking functions (gateways) is crucial for this purpose. Current implementations follow the TIPHON model, which decomposes gateways into SG, MGC and MG components that are potentially remote to each other and communicate over IP. Due to GSTN provider policies, SGs are used for hiding the internals of their SS7-based backbones, resulting in signalling translation being a two-step process: first, GSTN signalling is extracted at the (telco-operated) SG and sent to the MGC over SCTP (SIGTRAN), and secondly the (possibly third-party) MGC performs the translation (and vice versa, for the opposite direction).

This chapter has described a different, modular and more generalised approach based on LIFT, in which the signalling translation takes place at the SG, whereas the MGC has a supervisory role in the composite gateway. Both approaches have been implemented in the VIA testbed and their comparative evaluation reveals that they are equally suitable for large-scale IP Telephony, since they have similar performance characteristics and they are also compatible with the softswitch model (which essentially integrates the SG and the MGC), currently perceived as the main solution for VoIP scalability, as discussed in Chapter 3. An expansion of the pluggable media engine concept presented in Chapter 5 in the gateway space has been found to work equally well and thus increases the flexibility of implementation. The final piece of the large-scale IP Telephony puzzle, call routing, is addressed in Chapter 7.

CHAPTER 7

An IP Telephony

Call Routing Protocol

7.1 Overview

Call routing is one of the most important functions for the cohesion of a communications network [9] and, together with addressing, it also constitutes a critical scalability factor [75]. For IP Telephony, this process is conducted at both layer 3 (handled by IP), and layers 5-7 (the TCP/IP application layer), where a number of problems such as different signalling/media path existence, address translation and gateway location, need to be efficiently resolved for large-scale operation. The current IETF standard for IP Telephony call routing, TRIP, is based on BGP-4 and thus follows its general performance profile, including a nearly flat network organisation with sub-optimal scalability that does not allow the adoption of the protocol, as is, in a global IP Switched Telephone Network.

This chapter presents a new, hierarchical approach to IP Telephony call routing and, in particular, to the gateway location problem. An estimation of the evolution of the ISTN size (including upper bounds of component numbers) is provided, the necessity of an alternative solution is shown, and an enhancement to TRIP, the Hierarchical IP Telephony (HIT) call routing protocol, which greatly simplifies hierarchy formation by allowing only call routers to participate, is proposed and evaluated, with emphasis on its scalability characteristics. This way, the third of the three representative areas identified in Chapter 1 is investigated and the corresponding part of the hypothesis is accordingly supported.

7.2 Related Work

Routing is an essential procedure for all communication networks, be they circuit switched or packet switched, and thus a large amount of technical and scientific literature discussing related issues exists [293], [363]. The GSTN is still dominated by fixed-traffic hierarchical topologies, with a gradual shift towards more dynamic, bandwidth-on-demand arrangements ongoing since the early 1980s [9], [26]; during the same period, the rapid expansion of the Internet user base led to the development of the first dynamic routing algorithms for IP, on evolutions of which contemporary networks are still based [144], [167], [300]. These latter protocols are also involved in IP Telephony call routing, a problem that has been faced for at least a decade already [164]. More specifically, as seen in Chapter 3, VoIP call routing is a two-level process: at the network layer, the conventional IP routing algorithms are used, and they can be considered mature enough to focus research efforts on the other level, the application layer, where an additional mechanism, TRIP [314], has already been proposed by the IETF, as a partial solution.

TRIP is essentially a service location protocol that enables IP Telephony call routers (i.e., signalling servers, MGCs, softswitches, or similar devices) to select the most appropriate path to a signalling interworking function (i.e., a gateway), according to certain criteria, whenever a call is destined for a non-VoIP network (typically the GSTN). The protocol resulted from the merger of two earlier specifications (GLP [349] and TBGP [148]), is heavily based on BGP-4 [306], is independent of signalling standards such as H.323 or SIP, and is seen as a solution to the overall gateway location problem [315], [316]. A lightweight version of TRIP, TGREP [17] (originally called TRIP-lite), can be used for advertising the attributes and current status of a gateway to TRIP entities (location servers) operating in its vicinity, and certain restrictions (such as collocation) are implied about the topology of these entities with respect to call routers, as already discussed in Chapter 3.

Surprisingly, very little work has been published on TRIP, despite its finalisation in RFC 3219 as of January 2002 [314]. A partial (and already defunct) implementation exists in the public domain as a module of the Vovida VOCAL stack [U66], while a limited performance evaluation of the protocol has been conducted, using a basic network model of two ITADs and a correspondingly low number of SIP proxy server emulators to reach call blocking and rerouting suggestions that cannot easily be generalised [338]. Furthermore, no large-scale commercial IPTNs incorporate TRIP, despite its monopoly, as providers seem to prefer static configuration or proprietary solutions for gateway location, thus resulting in the creation of isolated "islands" of VoIP connectivity within the "ocean" of the GSTN. But even if the protocol were deployed globally, it can be doubted whether it would have been an adequate solution in terms of scalability, offered features and breadth of interoperability scenarios covered, since, as seen in Chapter 3 and Chapter 4, the IP Telephony call routing problem is not restricted to the IP-to-GSTN case, not to mention the possibility of more than one gateways being needed on the end-to-end path from source to destination. The requirement for an alternative solution will probably be exacerbated as IP Telephony coverage grows and, vice versa, a scalable solution will significantly accelerate that growth; however, no such solution has been researched thus far.

The almost complete lack of data on TRIP performance described above, can be partially offset by referring to the large amount of corresponding work on BGP-4 (mainly approximate assessments, since the complexity of inter-domain networking makes analytical modelling a very difficult task). As seen in Chapter 3, TRIP is largely a customisation of BGP-4 for IP

Telephony call routing, therefore it can be assumed that, given their common architecture (down to the finite state machine level), the two protocols will behave in comparable ways, in most cases. Several studies analysing the performance of BGP-4 exist [46], [94], [151], [157], [259], [262], [264], [354], [396], [400] and, although they vary in terms of the particular aspect of the protocol they investigate, a common theme often found is the issue of scalability, as the BGP-4 routing table, due to the constant expansion of the Internet, exhibits exponential growth (Figure 7.1) [168], [262], [U15], [U44], despite the introduction of techniques like Classless Inter-Domain Routing (CIDR) [119] that have partially alleviated the problem. Policy-based routing, another core property of BGP-4, also helps by limiting the number of available paths and thus reducing the routing table size, but it rather exacerbates the complexity of the protocol (particularly the configuration and the decision process) [94], [135].

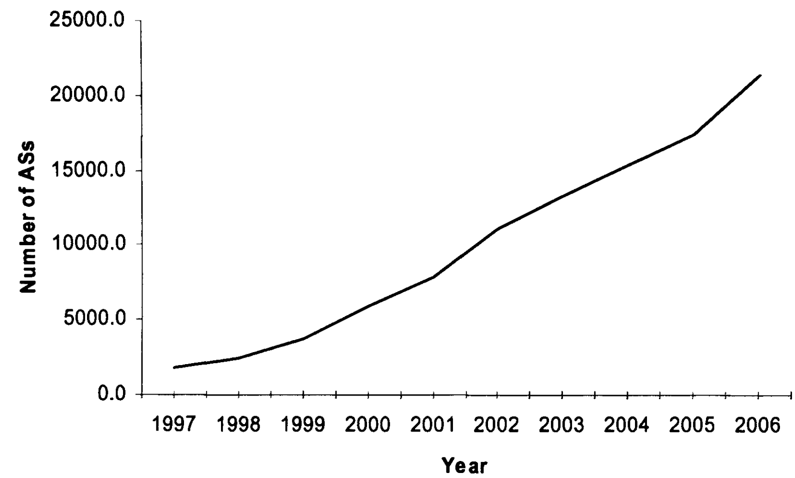


Figure 7.1: Growth of the BGP-4 routing table (announced ASs per year)

Property	Value
Routing Table Size (RIB)	669,783
Number of Networks	178,459
Number of ASs	21,803
Average AS Distance	3.5
Advertised IPv4 Addresses	1,501,588,935

Table 7.1: BGP-4 Statistics (as observed at AS1221and AS4637, on 08/04/2006) [U15]

The scalability of BGP-4 networks is further hampered by the need to establish a full logical mesh among the routers of each Autonomous System (i.e., for I-BGP), which requires $n(n-1)/2 = O(n^2)$ links (sessions) to interconnect *n* routers and, crucially, a reconfiguration of all these devices every time a new one is added or removed. Two mechanisms have been proposed for overcoming the problem, *route reflection* and *confederations*, both of which, interestingly, impose some sort of hierarchy on the normal (flat) operational model of the protocol (Figure 7.2). In route reflection, certain routers within an AS re-advertise ("reflect") selected routes to all other routers in the AS, without the need for a full mesh [22]; confederations, on the other hand, are adopted from the ISO Inter-Domain Routing Protocol (IDRP) [236], and can be used to group a number of smaller ASs (subdomains) into a larger, clustered one (superdomain), thus reducing the overall routing complexity, provided that efficient aggregation and route selection mechanisms also exist [371]. Although, contrary to BGP-4, TRIP does not require a full logical mesh among the LSs of an ITAD for its operation, it is also strained in large size networks and for inter-domain call routing; therefore, at least the confederations concept of BGP-4 might still be applicable, as will be shown later on. In either case, the nearly flat architecture used by both BGP-4 and TRIP constitutes one of the key factors for their growth deficiencies and has already been identified as such [264], [361].

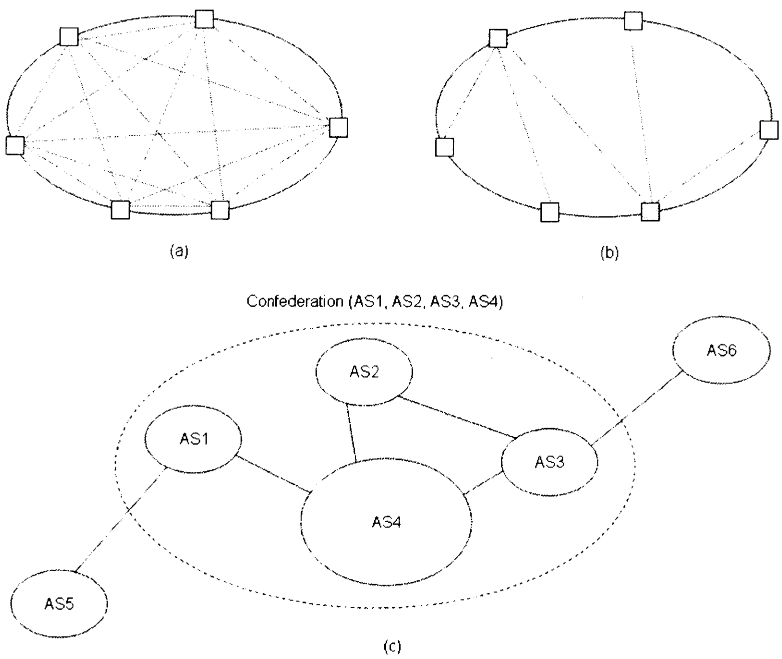


Figure 7.2: A BGP-4 (a) full mesh (b) route reflection and (c) confederation

Despite the scalability issues that will have to be addressed in the call routing area as IP Telephony expands, and the popularity of hierarchical solutions in other communication technologies, no similar suggestion can be found in the existing literature for VoIP. This chapter presents such a solution, the Hierarchical IP Telephony (HIT) protocol, an extension to TRIP that is more suitable for large-scale networks. HIT is a generic, LIFT-compliant call routing mechanism, which follows the guidelines set out in Chapter 4, where also the main differences with the TRIP model are analysed. More specific details about the technologies discussed in this section and the rest of the chapter can be found in the corresponding parts of the literature review, presented in Chapters 2 and 3.

7.3 Architecture and Design

The creation of a realistic network model for simulating IP Telephony traffic is a difficult task on its own, further complicated by the tendency of providers not to release statistics or topological details related to their networks. However, by complementing the little information available on the GSTN and mobile networks (including figures published in the March 2006 ITU worldwide telecommunications survey [170], summarised in Table 7.2), with a set or reasonable assumptions, such a model can be built, as discussed in the following text.

More specifically, in the rest of the chapter, general growth projections for the ISTN are provided according to already known data and usage evolution patterns, the scalability problems of the flat architecture employed by TRIP are identified, and a hierarchical solution is proposed instead. That solution is subsequently used for building a scalable network model, the parameters of which are associated with reasonable upper bounds, which are expressed in Big-O notation of integer powers of 10. A research implementation of this design is also included in the end, for evaluation purposes.

Category	Connections	Magnitude
IP Telephony	$0.4 * 10^9$	$O(10^9)$
GSTN	$1.2 * 10^9$	$O(10^9)$
Mobile	$1.8 * 10^9$	$O(10^9)$
TOTAL	$3.4 * 10^9$	$O(10^9)$

Table 7.2: Current number of voice connections (subscriptions) worldwide [170]

7.3.1 Estimated Network Size

An approximation of the future expansion of the ISTN (i.e., growth projections expressed in orders of magnitude) can be made for the number of terminals, gateways, call routers and ITADs of such a (nearly flat) network, using existing or estimated data, as in the following analysis, the results of which are summarised in Table 7.3.

Component	**Size**
Terminals	$O(10^{10})$
Gateways	$O(10^{6})$
Call Routers	$O(10^{6})$
ITADs	$O(10^{5})$
Terminals per ITAD	$O(10^{5})$
Call Routers per ITAD	$O(10^{2})$

Table 7.3: Estimated numbers of IP Telephony components for the year 2050

7.3.1.1 Terminals

A global IP Switched Telephone Network will have to encompass all existing switched (mainly GSTN and mobile) and packet (predominantly IP-based already) user communities. Assuming that each type of connection ("line") corresponds to 1 dedicated terminal (on average), the total number of terminals, *T*, will be approximately equal to the sum of GSTN, mobile and IP Telephony connections (denoted with *G*, *M* and *I*, respectively). Furthermore, if each Internet-connected host represents 1 IP Telephony connection, *T* can be closely approximated as

$$T = G + M + I$$

As seen in Table 7.2, the current total number of such connections (hence, networked terminals) is estimated by the ITU at about $T = 3.4 * 10^9$. The population of the planet, *P*, is almost double, at $P = 6.5 * 10^9$, and projected to reach $P' = 9.2 * 10^9$ by 2050 [U57]. Assuming that

- There will be a constant expansion of IP Telephony,
- New connections will continue to be GSTN, mobile or IP-based,
- An amount of overlap will exist (i.e., many users will choose one or two different connections, but not all three), smoothing out the bias caused by more technology-savvy users, who may choose to have connections of all 3 types, plus more than 1 per type (e.g., 2 mobile phones),

we can reasonably predict that the ratio R_{tp} of total connections (terminals) over total population

$$R_{tp} = T / P = (3.4 * 10^9) / (6.5 * 10^9) = 0.52$$

will continue to be close to $R_{tp}' = 1$, thus even in 2050 there will be approximately

$$T' = R_{tp}' * P' = 9.2 * 10^9 = O(10^{10})$$

total terminals in the global voice communications network.

The current number of IP Telephony terminals is about $I = 0.4 * 10^9$ (Table 7.2), used by about 10^9 users that are estimated to double by 2015 [U8]. Although no accurate predictions can be made for as far as 2050, it is reasonable to assume that this number will continuously grow, as Internet coverage spreads and mobile networks switch to fully IP-based communication. At the same time, the GSTN will grow much more slowly compared to VoIP [170]. Thus, the ratio of IP Telephony to switched (GSTN and mobile) networks which is currently

$$R_{is} = I / (G + M) = 0.13$$

could be biased towards a much greater value, i.e.

$$R_{is}' = (I + M) / G = 1.83$$

and probably reach even higher levels, since I is expected to continuously grow and M will further be increased as mobile telephony spreads (and progressively switches to IP), while G remains relatively constant [170]. The figures shown in Table 7.2 reveal that IP Telephony terminals currently represent about I/T = 11.8% of the total voice connections in operation, i.e. there is an approximate 90-10 split between switched and packet-based lines worldwide. Given the $O(10^{10})$ terminals estimated above for 2050 and the slow growth rate of the GSTN, this split will be reversed (even towards 10-90), if mobile networks have become fully IP-based by then. In any case, it is obvious that, out of the $O(10^{10})$ total terminals, a non-negligible fraction will continue to be connected to the circuit-switched GSTN (at least in the last mile, even if that is partially fibre-based) or a switched mobile network, while another (much larger) fraction will be IP-based. This means that each of these categories will be populated by $O(10^9)$ to $O(10^{10})$ terminals in the next few decades, thus $O(10^{10})$ is a fairly safe upper limit to assume for both.

7.3.1.2 Gateways

Given current and projected figures of scalability (in terms of support for simultaneous calls), LIFT gateways (from stand-alone signalling servers to complex, softswitch-controlled architectures) will probably be installed at digital local exchanges (DLEs) in order to implement VoIP within the GSTN, beyond the last mile, thus covering the GSTN-IP call scenario. However, in the opposite direction, IP-to-GSTN, the situation is more complicated. A 1998 study [315] estimates a need for about 18 * 10^3 gateways, based on the then number of Internet hosts (just 16 * 10^6) and approximate number of Points of Presence (POPs) in operation (such POPs could, for instance, be Digital Subscriber Line Access Multiplexers, DSLAMs, in xDSL networks); this figure implies about 1 gateway for every $O(10^3)$ hosts (terminals), or at least $O(10^6)$ gateways for the total of $O(10^9)$ to $O(10^{10})$ terminals estimated earlier on. Assuming a capacity of simultaneous calls ranging between $O(10^4)$ and $O(10^5)$ per softswitch (controlling multiple media gateways) [281], that figure is also consistent with the worst case scenario of all users placing a call at the same time, given that the number of users will be increasing towards $O(10^{10})$ in the next few decades, as already discussed.

7.3.1.3 Call Routers

IP Telephony call routers (i.e., TRIP location servers) will store and propagate gateway reachability information, that is, ranges of telephone number prefixes associated with the IPV4 addresses of the gateways that service them and the set of attributes (e.g., supported CODECs, capacity, or current load) valid on each route. While it is difficult to provide an exact estimation of the number of LSs required to efficiently cover the entire ISTN, it can be assumed from practical experience with BGP-4 [259] that, for each ITAD, there will be $O(10^1)$ to $O(10^2)$ such LSs. Furthermore, according to the TRIP operational model (analysed in Chapter 3), LSs will often be collocated with gateways (signalling servers), so the total number of LSs will be comparable to that of gateways, i.e. at least $O(10^6)$. The same figure can be derived from the number of ITADs, times the number of LSs per ITAD, as seen in the next section.

7.3.1.4 ITADs

BGP-4 uses a 16-bit field to identify Autonomous Systems, thus the maximum number of AS's in operation can be up to 65,536, but this is only a theoretical upper limit, since inefficiencies in the allocation of AS addresses may lead to the exhaustion of nearly half the available space, as is typically the case with binary addressing systems [28], [98], [165], [262]. The actual number of ASs is already more than 21,000 (Table 7.1) so there are proposals inside the IETF of using a 4-octet field for AS identification [379]. In either case, the number of BGP-4 ASs can be anticipated in the vicinity of $O(10^5)$. Assuming that TRIP, being heavily based on BGP-4, will exhibit a similar behaviour, and given its larger address space and richer set of attributes, we can anticipate the number of ITADs to be at least in the order of $O(10^5)$, too, for the full set of $O(10^{10})$ terminals that will be operating in the ISTN. Based on these figures, the number of terminals per ITAD can be expected to be $O(10^5)$, while the number of call routers per ITAD cannot exceed $O(10^2)$, as already estimated in the previous section.

7.3.2 A Hierarchical Approach

The estimated ISTN sizes and the scalability issues associated with BGP-4 and TRIP mentioned earlier on, point to the need for an improved call routing solution that will be able to accommodate the anticipated growth of VoIP in the near future. Indeed, several performance analysis studies recently published have raised concerns about the nearly flat network organisation that BGP-4 uses, which becomes increasingly inefficient as the network size grows [46], [157], [259], [262], [264], [354], [359], [396], [400]. The BGP-4 routing tables, for instance, are growing exponentially towards $O(10^6)$ entries (Table 7.1), and given the close similarity of TRIP with BGP-4, the same pattern can be expected for IP Telephony call routers. In such an environment, current (as well as forthcoming) hardware and software technologies would find it very difficult to cope with the corresponding processing load, no matter how much efficient the aggregation techniques employed would be; similar might become the situation with other ISTN entities (terminals, signalling servers, softswitches and gateways), which will have to be able to handle the large amount of messages generated by the variety of signalling and media protocols employed in IP Telephony calls.

The improved solution referred to above can be found among hierarchical architectures, which constitute a standard way for developing scalable networks, circuit-switched or packet switched, as already discussed [9], [11], [26], [62], [225], [361], [363]; actually, a hierarchical approach is not only attractive in scalability terms, but it could also prove to be the only viable alternative, at least for parts of the network (e.g., signalling, call routing or interworking functions). Such an architecture would be defined by several core parameters, including the number of hierarchy levels, the number of nodes per level, the method of hierarchy formation, the addressing scheme used and the call routing mechanism employed. For the case of IP Telephony, these parameters are examined in detail in the following sections and their approximated values are subsequently summarised in Table 7.4.

7.3.2.1 Hierarchy Levels

TRIP follows the BGP-4 model, which is based on a 3-level quasi-hierarchical network organisation, with *Hosts* (terminals) at level 0, *Networks* at level 1 and *Autonomous Systems* at level 2. The GSTN, on the other hand, follows a more hierarchical structure, with less than 10 =

$O(10^1)$ levels of hierarchy, so it can be used as a reference for analysing the possible structure of a hierarchical IP Telephony network.

In the United States, for example, there are up to six levels of hierarchy in the Bell telephone system, defined as (from bottom to top) [26]:

1. Terminals (Telephones)
2. Central Office (CO)
3. Toll Office
4. Primary Center
5. Sectional Center
6. Regional Center

Similarly, in Britain, the BT network is organised as follows (again, bottom to top) [U11]:

1. Terminals (Telephones)
2. Main Distribution Frame (MDF)
3. Remote Concentrator Unit (RCU)
4. Digital Local Exchange (DLE), potentially integrating MDFs and RCUs in the same physical cabinet
5. Tandem Switch (Digital Main Switching Unit, DMSU, or newer types of switches)

Even accounting for an additional international level, the total number of hierarchy levels in the telephony networks of those two countries, which have teledensities among the largest in the world [170], is less than 10. In another classic example of a hierarchical structure, PNNI-based ATM networks, the number of hierarchy levels can in theory be up to 105, but in practice it is anticipated that fewer than 10 will be needed to accommodate any size of national or international network [11], [258].

It is therefore reasonable to assume that, even for a global scale IPTN, like the ISTN, no more than $O(10^1)$ levels of hierarchy will be needed for efficient operation.

7.3.2.2 Nodes per Hierarchy Level

Hierarchies, by definition, are built by scaling core network parameters. In the United Kingdom, for instance, the BT telephony network consists of [U11]:

- Tens of millions of terminals (i.e., less than $100{,}000{,}000 = 10^8$)
- Tens of thousands of MDFs (i.e., less than $100{,}000 = 10^5$)
- Around 7,000 RCUs (i.e., less than $10{,}000 = 10^4$)
- Around 800 DLEs (i.e., less than $1{,}000 = 10^3$)

- Around 80 DMSUs plus a few other tandem switches (i.e., in the order of $100 = 10^2$)

As far as the terminals are concerned, a typical RCU serves a few thousand terminals (i.e., there are usually less than $10{,}000 = 10^4$ terminals per RCU), which agrees with the total number of RCUs and terminals (i.e., less than $10^4 * 10^4 = 10^8$) [U11].

These figures allow a more detailed analysis of the hierarchy of the BT network (or any similar infrastructure) and the calculation of maximum values (upper bounds) of nodes for each level. More specifically, based on the above discussion, it can be said for the number of terminals, T_1, per level-1 switching unit (MDF) and the number of terminals, T_2, per level-2 switching unit (RCU), that:

$$T_1 \leq 10^3 \qquad (1a)$$

$$T_2 \leq 10^4 \qquad (1b)$$

As with all hierarchies, there is a clear tendency to reduce the number of entities per switch level by an order of magnitude (i.e., 10). That provides a way of estimating the number of nodes (i.e., terminals for level 0 and switches for higher levels), S_i, per level (height) i, for an $(n+1)$-level hierarchy with an order of magnitude (ratio) m, having a nodes at the top:

- At level n: $S_n = a = a\,m^0$
- At level n-1: $S_{n-1} = a\,m = a\,m^1$
- At level n-2: $S_{n-2} = a\,m\,m = a\,m^2$
- . . .
- At level n-i: $S_{n-i} = a\,m^i = a\,m^{n-(n-i)},\ 0 \leq i \leq n-1$
- . . .
- At level i: $S_i = a\,m^{n-i},\ 1 \leq i \leq n$
- . . .
- At level 1: $S_1 = a\,m^{n-1}$
- At level 0: $S_0 = T_1\,S_1 = T_1\,a\,m^{n-1}$ (2)

Equation (2) provides an estimation of the number of nodes at level 0 (i.e., the terminals) in the network. Although the number of these terminals does not follow the approximate geometric pattern observed at the other levels, one can reasonably assume that, with improved capacity level 1 switches, a significant (e.g., tenfold) increase in the terminal number can be sustained without invalidating the model for the higher levels. Indeed, contemporary media gateways can serve at least $O(10^4)$ terminals, as already discussed.

Based on the above and also noting that a, m and n are integers which have to accommodate the entire size of the network model (in other words, they have to be rounded up), the values of those parameters can be estimated. More specifically, solving Equation (2) for a, m and n, produces [217]

$$a = \left\lceil \frac{S_0}{m^{n-1}T_1} \right\rceil = \left\lceil \frac{S_1}{m^{n-1}} \right\rceil \quad (3)$$

$$m = \left\lceil \sqrt[n-1]{\frac{S_0}{aT_1}} \right\rceil = \left\lceil \sqrt[n-1]{\frac{S_1}{a}} \right\rceil \quad (4)$$

$$n = 1 + \left\lceil \log_m\left(\frac{S_0}{aT_1}\right) \right\rceil = 1 + \left\lceil \log_m\left(\frac{S_1}{a}\right) \right\rceil \qquad (5)$$

The above approximations allow an estimation of the number of top-level switches, a, and, eventually, the total number of switches at level i, S_i, if the number of switches at level 1, S_1, and the level-by-level ratio (scaling factor), m, are known. A nice attribute of equations (3)-(5) is that they are not dependent on the value of T_1; in other words, any number of level-0 hosts (e.g., terminals) can be accommodated in the model (within reasonable resource restrictions).

So, in a network that follows a hierarchical organisation pattern of *(n+1)* levels with scaling factor m, at height h ($1 \le h \le n$) the maximum total number of nodes, S_h, will be

$$S_h = a\, m^{n-h} \qquad (6)$$

where $a \le O(10^2)$ and $m > 1$ for any efficient hierarchy. It is obvious that, progressing geometrically like above, for n+1 levels of hierarchy, where terminals reside at level 0 and switching nodes at levels 1 to n, the maximum total number of switching nodes (i.e., excluding the terminals) will be [217]

$$\sigma = \sum_{k=1}^{n} am^{n-k} = \sum_{k=1}^{n} am^{k-1} \Rightarrow$$

$$\sigma = \frac{a(m^n - 1)}{m-1} = O(m^{n-1}) \quad (7)$$

For the actual BT network figures quoted earlier on, the observed pattern is

$$a=100,\ m=10,\ n=4 \qquad (8)$$

which, in turn, gives a maximum total number of IP Telephony switching nodes of

$$\sigma = 100*(10^4-1) / (10-1) = 111{,}100$$

If the order of magnitude were 2, then the corresponding number would have been

$$\sigma = 100*(2^4-1) / (2-1) = 1{,}500$$

To support the same number of terminals, S_0, according to (2), T1 would have to be substantially increased, given the specific combination of a, m and n that results in $\sigma = 1{,}500$

above. Of course, this is a maximum number, which does not have to be followed completely, i.e., providers may manage to support the same number of level 0 hosts (terminals) by using a far smaller number of switching nodes in total.

7.3.2.3 Hierarchy Formation

All nodes in a hierarchical network are organised, for routing purposes, into multiple levels of aggregated groups. In hierarchical IPTNs, network components (terminals, gateways and call routers) residing at level 0 are grouped in ITADs at level 1, which can be progressively aggregated towards the top into higher level ITADs, according to a manual (e.g., static provider setup) or an automated (e.g., similar to RSPL [2] or to PNNI [10]) hierarchy formation mechanism. The overall architecture should be such as to support global connectivity for all communication scenarios identified in Chapter 6, as opposed to the IP-to-GSTN bias of TRIP.

7.3.2.4 Addressing

As mandated by LIFT, either IPv4 or IPv6 addresses can be used for the ISTN. IPv4 is the obvious candidate, due to its universal implementation and field-proven operation, and obviously all IP Telephony networks will have to be based on it for the near future.

However, the ongoing depletion of the IPv4 address space will, inevitably, lead to the universal adoption of IPv6. In such a case, the 128-bit (16-byte) IPv6 address field [90] can also be used to directly convey phone numbers, given a suitable prefix, which would denote that a particular address is actually an E.164 identifier. Assuming a maximum of 15 digits per telephone number [172] and a maximum of 1 nibble (4 bits) per decimal digit (to accommodate values 0-9), this would require 4x15=60 bits (7.5 bytes), for "direct" translation. For example, the London telephone number 0044-20-7387 1397 would become, in an 8-byte format:

```
xxxx xxxx | 0000 0000 | 0100 0100 | 0010 0000 |
0111 0011 | 1000 0111 | 0001 0011 | 1001 0111
```

The method proposed above uses a Binary-Coded-Decimal (BCD) notation that could be adopted as a generalised E.164 to IPv6 mapping. For example, the lower 8 bytes of the 16-byte IPv6 address field can be allocated for IP Telephony addresses, with the highest 4 bits of the 8th least significant byte used as a selector for various roles (e.g., E.164 address, IPv6 address, etc.). A single 64-bit prefix (constructed from the higher 8 bytes of the 16-byte IPv6 address) could then be used to uniquely denote the telephone number address class.

Some special cases merit further consideration. According to the IPv6 specification [90], the last 10 bytes (2 bytes SLA + 8 bytes Interface ID) are constant, even when a site is renumbered, thus providing 2x10 = 20 nibbles for BCD-based identification of up to 19 phone number digits (assuming that the highest nibble will be used as a selector). There can be non-E.164 numbers longer than that, or special needs for IP Telephony (e.g., carrying a port number, which essentially is similar to a telephone extension, but maybe the caller would like it included in the original address instead of waiting for the signalling protocol to do the negotiation).

In any case, apart from the first two bytes of the IPv6 address (accommodating for both unicast and multicast transmission), the rest could be used in a format "C::id", where *C* represents the application class (e.g., IP Telephony) for which the 16-byte address is used and *id* denotes the actual identifier of the device addressed. So, for example, by reserving a 2-byte prefix for IP Telephony as a "class of application", that gives $2^{2*8} = 2^{16} = 65{,}536$ potential different classes of applications and allowance for up to 2*(16-2-2) = 24-digit telephone numbers, which is more than enough, even for the special cases mentioned above. IPv6 also reserves a particular prefix using the first 57 bits of the 64-bit Interface ID (1111111011111...111) for anycast communication, but that does not represent any valid phone number according to the BCD notation, so it poses no problem for the method suggested.

IPv6 addressing, though, has its own efficiency problems. The large size (16 bytes) of the address field translates into longer overheads, i.e. higher bandwidth consumption for carrying the same amount of multimedia information (e.g., voice). Most importantly, the vast increase in the available address space can lead to higher fragmentation and a significantly larger number of subnets, thus further loading the routing tables of inter-domain protocols like BGP-4 and TRIP.

Irrespective of the network layer mechanism (IPv4 or IPv6), the key assumption for a hierarchical routing scheme to work efficiently is that addresses are highly aggregatable, so that multiple levels of the hierarchy can be formed. The structure inherent in IPv4 addresses is helpful for constructing hierarchies of a few (3 to 4) levels. IPv6 addresses obviously excel in this area, too, and the BCD format proposed above allows an aggregation process parallel to that of the telephony network, which, as already discussed, is hierarchical with up to $O(10^1)$ levels above level 0 and this parallelism greatly simplifies the formation of higher-level ITADs.

As far at it concerns higher levels of the hierarchy (i.e., those above level 0), addressing could be in the form A.B.C[.D], where, similarly to PNNI [11]

- A denotes the level of hierarchy,
- B denotes the ITAD,
- C denotes the (potentially aggregated) node within the ITAD,
- D optionally denotes the children of the ITAD node, if the node is an aggregated one.

7.3.2.5 Call Routing

The proposed hierarchical organisation of the network will affect the actual IP Telephony call routing process at both the application layer and the network layer. However, in the latter case, communication will remain fully compliant to the established TCP/IP model, therefore IP can be assumed to handle the necessary operations, so as to focus on the effects at the application layer. Even so, as specified by LIFT, a number of options exist, including the decision whether to search for both a signalling and a media path, the type of routing (hop-by-hop or source) to be employed, the route advertisement method and scope pursued, the kind of connectivity offered (local or global) and the communication scenarios supported. All these issues are directly related to the actual call routing protocol employed in the network.

Component	Size
Hierarchy Levels	$O(10^1)$
Nodes (Call Routers) per Hierarchy Level *i*	$S_i = a\, m^{n-i},\ 1 \le i \le n$
Address Space	$O(10^{10})$

Table 7.4: Maximum projected values for the components of a hierarchical IPTN

7.3.3 The Hierarchical IP Telephony (HIT) Protocol

HIT is a generic call routing mechanism, which follows the hierarchical architecture analysed in the previous section. More specifically, the protocol aims at achieving large-scale IP Telephony connectivity by employing a hierarchical organisation of the network and loose source routing. An outline of the main functions proposed by HIT is presented in the following sections.

7.3.3.1 Network Organisation

In the HIT model, only the call routers (e.g., TRIP location servers) participate in the hierarchy, whereas the rest of the network is organised as a flat structure. This architecture maintains maximum possible compatibility with the conventional TRIP model, it can achieve faster hierarchy formation and reconfiguration times due to the smaller number of nodes involved, and, in conjunction with source routing, it solves the problem of identifying the address of a gateway without affecting the operation of the network. More specifically, both the signalling path and the media path remain disjoint and the only addition is the intervention of one (or, in rarer cases, more) gateways on either, as needed for protocol translation purposes.

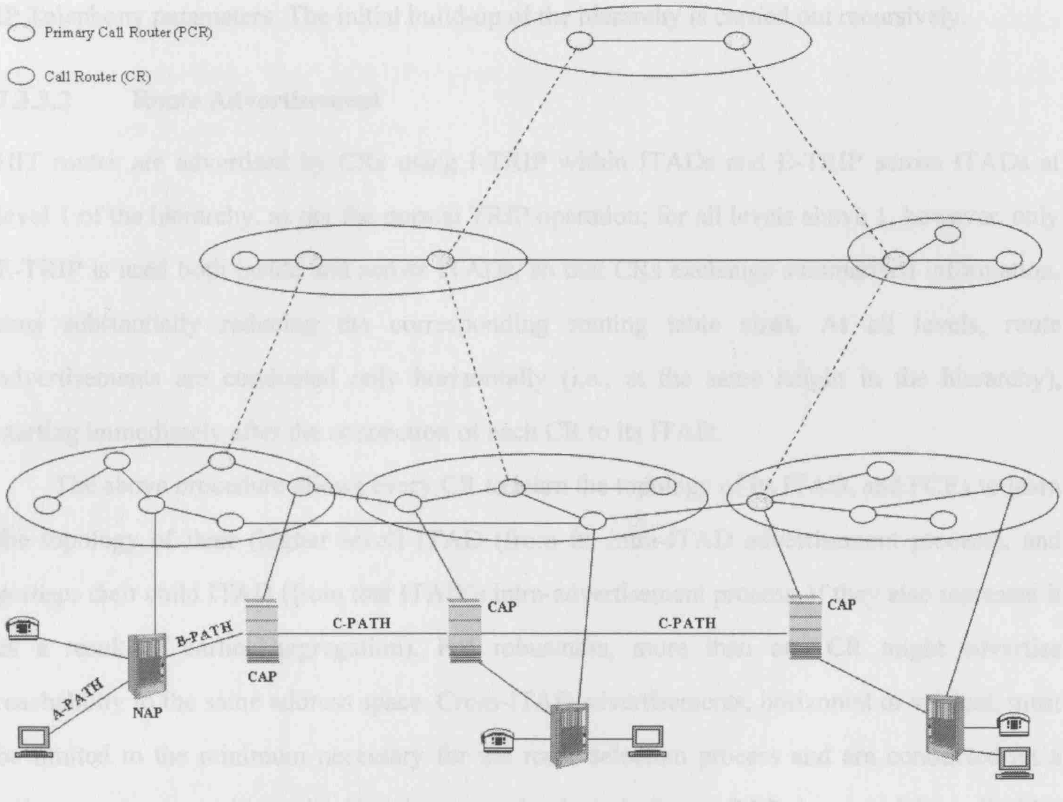


Figure 7.3: HIT network organisation

Call routers in the ISTN are organised into multiple levels of aggregated groups (ITADs), each of which is created and placed in the network hierarchy via administrative action (e.g., in accordance with provider policies or Service Level Agreements). In every ITAD, one of the resident call routers (CRs) is designated as the Primary Call Router (PCR), much like the Peer Group Leader (PGL) in PNNI [11]. Every CR in an ITAD can learn about the PCR(s) of that ITAD via a broadcast or flooding mechanism. Hierarchy changes are periodically announced in the network via PCRs. After aggregation, each ITAD is represented by its PCR at the higher level ITAD. For scaling purposes, the network should be engineered in such a way that, at each aggregated ITAD, the machine elected as a Call Router is not already a PCR for a lower level domain. A simplified view of the network organisation that results from applying the above techniques to the LIFT model presented in Chapter 4, is depicted in Figure 7.3.

The hierarchical organisation of the network is represented in HIT by a tree data structure [225], to achieve fast route search and quick dynamic operations (such as prune and graft). All the nodes of the tree, including the leaves, are ITADs; therefore, given a tree of degree d, organising N "physical" (non-aggregated) ITAD's, up to $log_d N$ levels are required. The cost associated with each tree branch can be calculated from a weighted function with a number of IP Telephony parameters. The initial build-up of the hierarchy is carried out recursively.

7.3.3.2 Route Advertisement

HIT routes are advertised by CRs using I-TRIP within ITADs and E-TRIP across ITADs at level 1 of the hierarchy, as per the normal TRIP operation; for all levels above 1, however, only E-TRIP is used both inside and across ITADs, so that CRs exchange summarised information, thus substantially reducing the corresponding routing table sizes. At all levels, route advertisements are conducted only horizontally (i.e., at the same height in the hierarchy), starting immediately after the connection of each CR to its ITAD.

The above procedure allows every CR to learn the topology of its ITAD, and PCRs to learn the topology of their (higher level) ITAD (from its intra-ITAD advertisement process), and perhaps their child ITAD (from that ITAD's intra-advertisement process, if they also represent it as a result of earlier aggregation). For robustness, more than one CR might advertise reachability to the same address space. Cross-ITAD advertisements, horizontal or vertical, must be limited to the minimum necessary for the route selection process and are conducted on a point-to-point or multicast basis, using a mechanism similar to BGP-4, a special (replicable) registration server, or anycast communication.

7.3.3.3 Route Selection

Route selection (gateway discovery) in HIT is necessarily a multilevel process, due to the hierarchical organisation of the network and the need to discover an end-to-end path, because of the source routing model followed. Figure 7.4 summarises the procedure.

To establish a call, an end user terminal contacts its local gateway (Network Access Point, NAP), which can be anything from a stand-alone signalling server to a large-scale softswitch. If the NAP cannot connect to the destination directly (i.e., if the destination is not one of the terminals that this NAP services), through its CAP it contacts a CR, the address of which it knows via static configuration or an ARP-like mechanism. The CR is then responsible for finding an end-to-end path (which might only consist of one LIFT gateway, but it can also include more, if necessary). If the CR cannot decide an end-to-end path (e.g., because the destination lies outside the CR's ITAD, or possibly because it has not yet synchronised its route database with the PCR of the ITAD), the Primary Call Router of the ITAD is contacted. If the PCR cannot decide for a path, it contacts the PCR of its upper level domain. The process is repeated until, at a level of the hierarchy (e.g., of height n), an end-to-end path ("n-path") of one or more LIFT gateways has been discovered. This path is then sent directly to the CAP and the NAP of the calling terminal, so that the call can be established; otherwise, if at no level of the hierarchy a path can be found (e.g., if no such path exists), the NAP of the requesting terminal is notified accordingly.

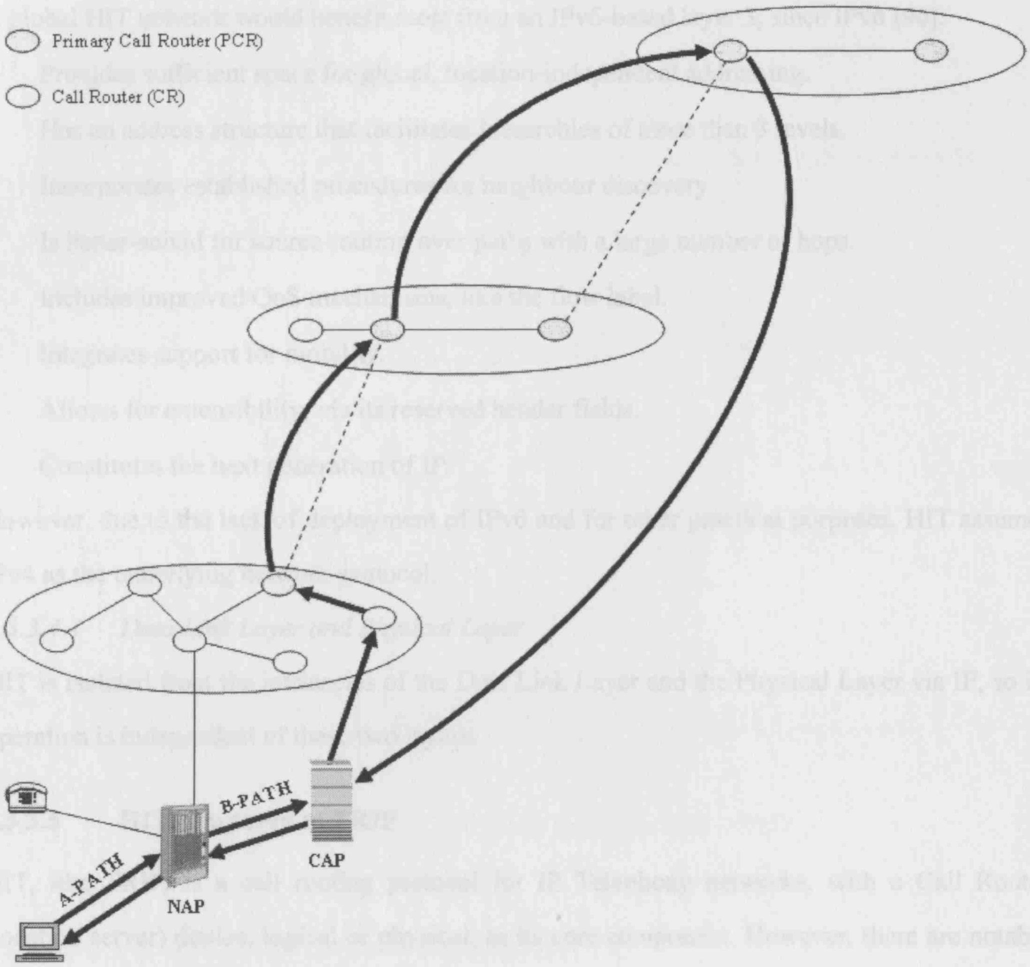


Figure 7.4: The HIT route selection process

7.3.3.4 HIT in the TCP/IP Protocol Stack

HIT makes certain assumptions about underlying protocols or communication mechanisms in the TCP/IP protocol stack.

7.3.3.4.1 Application Layer

HIT operates at the application layer. It is independent of other application layer protocols (e.g., SIP, H.323, RTP) used in the ISTN, and of the actual network layer topology of the IPTNs that comprise it.

7.3.3.4.2 Transport Layer

HIT assumes a reliable transport layer for its operation. This reliability should preferably be implemented via TCP or SCTP virtual channels, but UDP connections with a retransmission mechanism may also be used for increased efficiency.

7.3.3.4.3 Network Layer

A global HIT network would benefit most from an IPv6-based layer 3, since IPv6 [90]:

- Provides sufficient space for global, location-independent addressing.
- Has an address structure that facilitates hierarchies of more than 3 levels.
- Incorporates established procedures for neighbour discovery.
- Is better-suited for source routing over paths with a large number of hops.
- Includes improved QoS mechanisms, like the flow label.
- Integrates support for mobility.
- Allows for extensibility, via its reserved header fields.
- Constitutes the next generation of IP.

However, due to the lack of deployment of IPv6 and for other practical purposes, HIT assumes IPv4 as the underlying network protocol.

7.3.3.4.4 Data Link Layer and Physical Layer

HIT is isolated from the intricacies of the Data Link Layer and the Physical Layer via IP, so its operation is independent of these two layers.

7.3.3.5 HIT compared to TRIP

HIT, like TRIP, is a call routing protocol for IP Telephony networks, with a Call Router (location server) device, logical or physical, as its core component. However, there are notable differences between the HIT and TRIP architectures and operational models.

More specifically, a comparison between the two protocols reveals that:

(a) HIT aims at achieving global connectivity and is suitable for all call scenarios, not only the VoIP-to-GSTN one assumed by TRIP.

(b) HIT assumes a different view for the basic IPTN components (e.g., the gateway), through its LIFT compliance.

(c) HIT uses a multilevel hierarchical network organisation, as opposed to the nearly flat (up to three levels high) architecture than TRIP follows.

(d) HIT does not only discover the SCN exit to an IP network, but it also calculates a whole end-to-end path.

(e) HIT is a source routing protocol, not a hop-by-hop protocol like TRIP.

The comparison above reflects a different philosophy regarding the nature and the scope of a call routing protocol, and indicates the superiority of HIT for the implementation of large-scale IP Telephony networks.

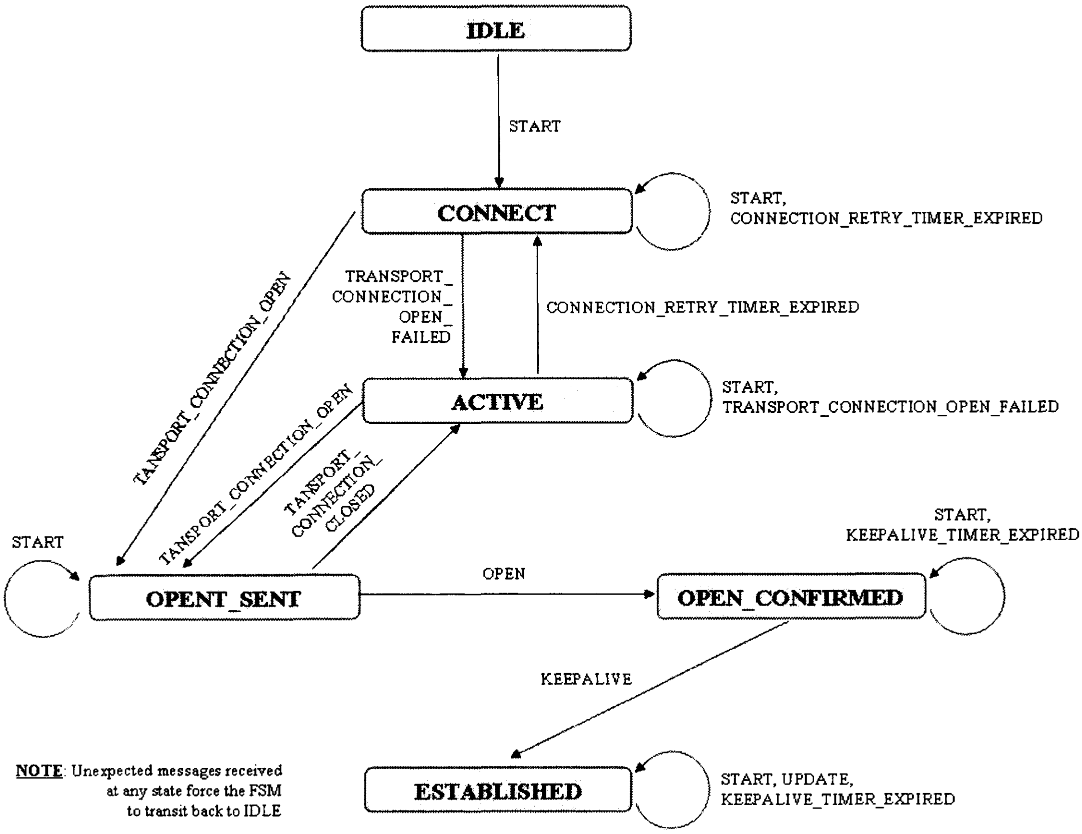


Figure 7.5: The TRIP Finite State Machine (FSM)

7.4 A Research Implementation

An instance of the architecture analysed in the previous section has been implemented, as a proof of concept, into a simple TRIP location server with extensions for hierarchical operation. The application developed is called the Hierarchical IP Telephony Routing Agent (HITRA) and its main characteristics are:

- **Native object-oriented design** around well-defined APIs, to facilitate extensibility, maintenance and portability.
- **Full TRIP implementation**, including the Finite State Machine (FSM), message formats, processing rules, routing tables and timers.

- **Hierarchical Extensions to TRIP**, to support large-scale call routing in current and future IP Telephony networks.
- **Event-driven operation**, for all components of the application.
- **Platform independence**, based on Java and the open specification of the TRIP and HIT protocols.

More details of the above characteristics and their actual implementation into HITRA are provided in the following sections.

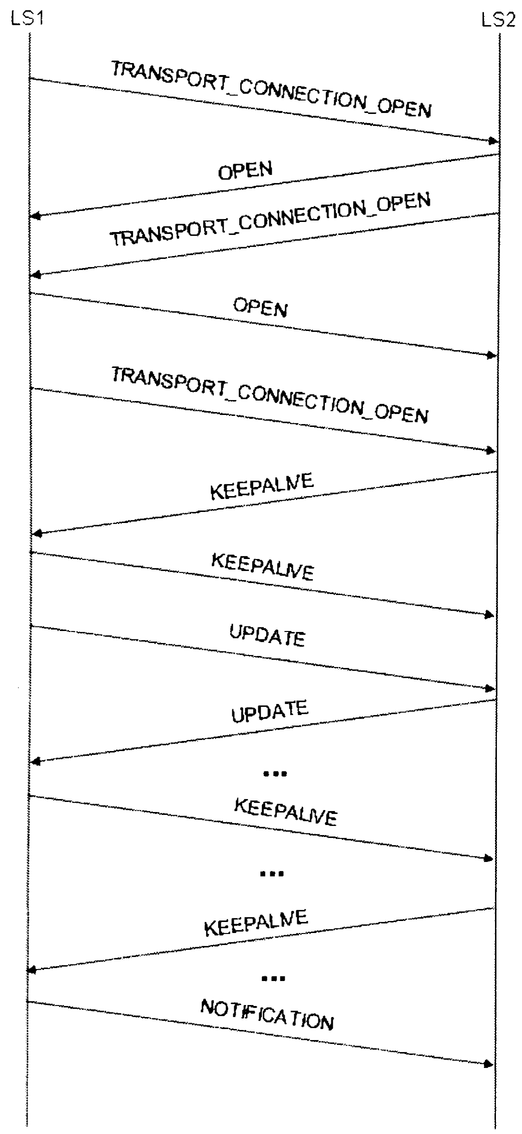


Figure 7.6: Message exchange between two external TRIP peers

7.4.1 TRIP in HITRA

HITRA implements the full TRIP Finite State Machine (Figure 7.5), as adopted from BGP-4

[306]. There are 6 states and 13 events (messages) specified by TRIP in RFC 3219 [314], each of which is represented in HITRA using a separate Java class, while an additional one, TRIPState, maintains and processes current state information in the application. To accommodate different message priorities, incoming messages are received in a custom weighted buffer queue, based on a Java ArrayList for faster performance. All message formats are supported as specified in RFC 3219, and all well-known message attributes are implemented in HITRA, along with their corresponding processing rules. The most frequently encountered message is UPDATE, used by location servers for periodically synchronising their routing tables (databases) with each other.

An overview of the initial message exchange that takes place between two external TRIP peers is depicted in Figure 7.6. There, two Locations Servers, LS1 and LS2, belonging to different ITADs (i.e., being external peers), start their communication after the succesful establishment of a unidirectional transport connection for both sides. The establishment of each connection is confirmed by a KEEPALIVE message, which causes the recipient LS to enter the ESTABLISHED state, in which it can send an UPDATE to its peer (and vice versa), thus effecting the synchronisation of the corresponding routing tables. The peering session is maintained via the periodic transmission of KEEPALIVE messages, until it is terminated by any of the two LSs, via a NOTIFICATION message.

Similarly to BGP-4, TRIP maintains a number of routing databases (tables), called Telephony Routing Information Bases (TRIBs). These vary according to the direction (inbound, outbound) and the origin (internal, external) of the routing messages stored. The routing table of each LS, Loc-TRIB, is populated by the internal (intra-domain) and the external (inter-domain) routes that have been selected from the set of all known routes according to the policies (i.e., the decision process) prevailing at the LS. A subset of these routes is advertised to other peer LSs in different ITADs. Routes are generally in the form **`<address_family, address_prefix, protocol, attributes>`**, where `address_family` can be E.164, DECIMAL or PENTADECIMAL, `address_prefix` represents the set of destination addresses reachable, the `protocol` can be H.323 or SIP and numerous `attributes` can be used, including a `NextHopServer` one, which records the address of the signalling server (e.g., MGC or softswitch) towards which control packets must be forwarded. Only the E.164 address space is supported by HITRA, thus the `address_prefix` parameter holds ranges of telephone number prefixes serviced by the corresponding gateway (the address of the MGC of which is propagated via the `NextHopServer` parameter). For optimal pattern matching of the address prefixes, a *Trie* data structure is used, which is essentially an ordered tree that stores an associative array of string keys in its nodes, indexed by the branches [261].

TRIP also specifies seven timers, which are all implemented in HITRA and configurable from within the application, as mandated by RFC 3219. The expiration of any of these timers results in the generation of a single HITRA event, represented by the TimeOut class, which carries identification information pointing to the cause of the timeout and its source, in order to facilitate the event handling process. Similarly, a single class, Timer, is used to represent all TRIP timers in HITRA, and an additional one, EventTimer, randomises the initialisation process to avoid synchronisation of LSs. The duration (expiration interval) of the 7 TRIP timers ranges from 10 seconds to 180 seconds [314].

7.4.2 Extensions to TRIP

TRIP assumes a nearly flat network architecture, which, as already discussed, imposes significant scalability restrictions. The solution proposed herein employs a hierarchical organisation, therefore certain extensions to the original specification are needed. HITRA introduces such extensions with the primary design goal of applying the minimum possible amount of modifications to the protocol, so that most of its field proven (via BGP-4) characteristics remain intact.

The mechanism used to represent a hierarchy of location servers in HITRA, is based on the *confederations* model of BGP-4 [306], [371]. Similarly to that model, a TRIP confederation is defined as a collection of ITADs ("member" ITADs) that is advertised using a special ITAD number (the ITAD Confederation Identifier, ICID), externally visible to location servers that do not belong to any of the ITADs of the confederation. All LSs that participate in the confederation use their member ITAD number for the communication with their peers in the confederation (internal LSs) and the ICID for all other peers (external LSs); otherwise, LSs handle ICIDs like normal ITAD numbers, stored in the "My ITAD" field of the TRIP OPEN message. An additional modification to the normal operation of TRIP is required for the particular attribute (RoutedPath) that records the list of ITADs encountered on the path to the destination, as happens with the AS_PATH attribute in the case of BGP-4. Numerous other, minor changes, are also implemented in HITRA, along the lines of RFC 3065 [371]. A top-level architecture of TRIP ITAD confederations is depicted in Figure 7.7.

A simpler technique can also be used for supporting hierarchical call routing in a TRIP-based IPTN. More specifically, instead of modifying certain parts of the protocol to implement the BGP-4 confederation mechanism, an addressing scheme of the form A.B.C[.D], as presented earlier on, can be employed for identifying the call routers without changing the routing tables (which only store telephone number prefixes, gateway addresses and attributes, as already seen), and the decision process can be adapted accordingly. Assuming that address allocation is conducted in a way so that high levels of aggregation can be achieved, this technique, although more proprietary, should work equally well to confederations, since it implicitly incorporates the hierarchical organisation of the network.

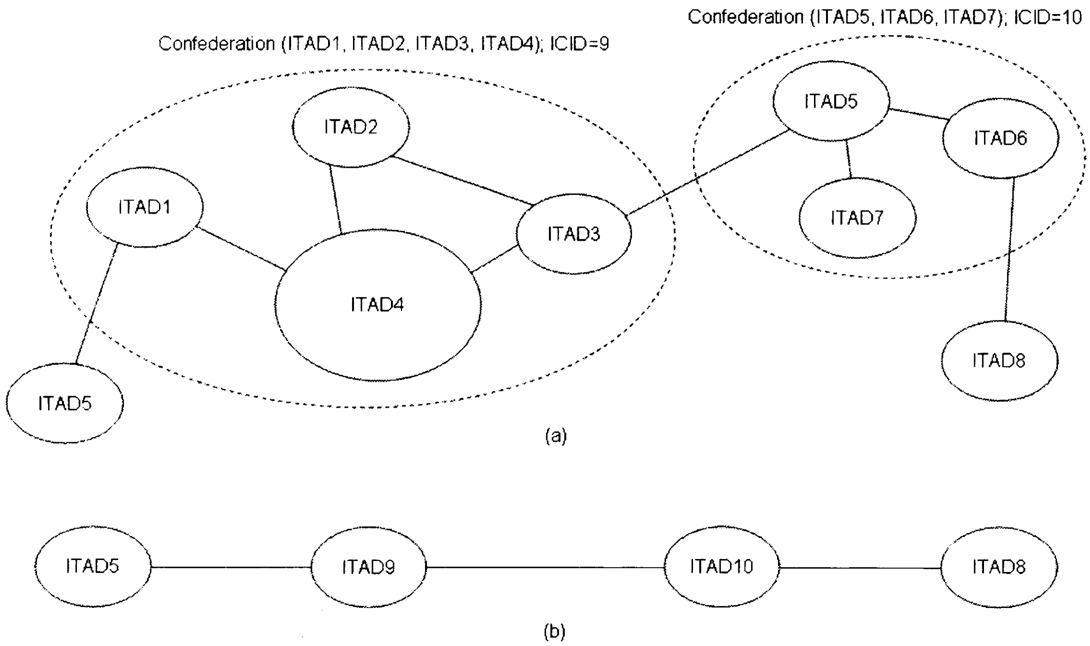


Figure 7.7: TRIP confederation view (a) before and (b) after aggregation

7.4.3 HITRA Operation

The HIT Routing Agent has a simple command-line interface, from which the base Java class, HITRA, is launched, sets numerous local and debugging variables and creates a TRIPSession object, which undertakes all call routing operations from then on. After its instantiation, the TRIPSession object initialises the data structures, the event handler and the network interfaces (TCP sockets) of the agent, and waits for incoming connections. No special security checks are performed in this prototype application, thus all such connections are, by default, accepted. Communication is conducted via persistent connections over a reliable transport mechanism (TCP). For every accepted connection, a new copy of the TRIP FSM is launched, which is constantly fed with messages by the event handler and processes them according to the rules specified in RFC 3219 [314]. The initial state of each FSM is IDLE and the FSM reverts back to that state if an error (e.g., connection reset, or reception of an irrelevant message) takes place; otherwise, a transition is progressively made through CONNECT, ACTIVE, OPEN_SENT and OPEN_CONFIRM, until finally the ESTABLISHED state is reached (Figure 7.5), at which the two LSs synchronise their databases.

Database synchronisation varies according to the scope of communication. In I-TRIP, a flooding mechanism similar to that used by OSPF [275] is invoked, in which an LS receiving an UPDATE message from any of its internal peers (i.e., those belonging to the same ITAD) forwards it to all other LSs it is currently connected, unless it has been the originator of the message (otherwise, a packet storm could be caused). In E-TRIP, on the other hand, LSs synchronise their databases by exchanging UPDATE messages in a peer-to-peer fashion, without further forwarding. The role of an LS can also be played by TGREP [17] clients that advertise the capabilities of the gateway where they are hosted, similarly to the MGC of the VIA gateway presented in Chapter 6. Each LS stores the unprocessed routes received from every other LS in the same ITAD in a separate database (Adj-TRIB-In) and maintains a single additional database (Ext-TRIB) for all routes received from external LSs. The routes stored in the Adj-TRIBs-In and in the Ext-TRIB of an LS, are processed by the LS according to its local policies as part of the routing decision process, and a subset is stored in a separate database (Adj-TRIB-Out) for every external peer to which the LS is currently connected.

Route advertisement and route selection take place as specified by the HIT protocol described earlier on, using both I-TRIP and E-TRIP. As discussed in Chapter 3, TRIP assumes that each location server will be co-located with a Signalling Server (e.g., a SIP Proxy Server, an MGC or even a softswitch), which, although restrictive in network topology terms, it nevertheless facilitates source route calculations; this is because the collocation implies a model where the vast majority (if not all) of LSs are hosted in gateways, thus knowing the corresponding routes already at their startup (via configuration), without the need for lengthy UPDATE message exchanges.

The full HITRA code structure is listed in Appendix B.

7.5 Evaluation

HITRA is a prototype implementation of TRIP and HIT, with emphasis on improving a core scalability characteristic of IP Telephony, call routing table size, which, as already seen, may be reaching the limits of current hardware and software technologies very soon. Like with all hierarchical network architectures, HIT significantly reduces space requirements by effectively distributing the overall routing table across different levels and performing address aggregation. This improvement also directly affects convergence time and it can be expressed in quantitative terms, using existing BGP-4 data.

Due to the scale of the problem addressed by HITRA, a mathematical analysis is bound to be more accurate than an implementation-based one and, for that purpose, BGP-4 can be used as the starting point. More specifically, the latest analysis of BGP-4 [264] estimates that the memory requirements (*MR*) of the protocol per Routing Information Base (RIB) maintained in an external peer (i.e., for E-BGP), depend on the average number (*N*) of reachable networks advertised by the peer, the mean length (*M*, counted in Autonomous Systems) of Internet paths, the total number (*A*) of ASs in operation, and the average number (*P*) of peers through which an AS is reachable. If *R* is the number of bytes needed to store each route advertisement and *S* the number of bytes needed to identify an AS, then the equation approximating the maximum value of *MR* (in bytes) is [264]

$$MR = [(N*R) + (M*A)*S] * P \quad (9)$$

The mean AS distance (*M*) grows slowly, compared to *N* and *A*, and is assumed to be close to 5 herein, although actual observations indicate that it is 3.5 (Table 7.1) and values of 10 or more have been suggested [264]). Given that R=4 and S=2 in BGP-4, equation (9) becomes

$$MR = [(N*R) + (M*A)*S] * P = (4*N + 10*A) * P \quad (10)$$

A graphical representation of (10) is shown in Figure 7.8, for different values of *N* (up to 1,000,000) and *A* (up to 50,000), assuming P=10. As the size of the routing table increases linearly with *P*, for P=100 (also considered a plausible scenario [264]), the memory requirements will be tenfold.

Based on (9), the number of individual entries, *E*, in the BGP-4 routing table (RIB) will be

$$E = O((N + (M*A))*P) \quad (11)$$

Clearly the dominant factor in equation (11) is *N* and, as seen in Figure 7.8, beyond 200,000 networks, the routing table size becomes almost linear to *N*. With a current value of around 180,000 = $O(10^5)$ [264], [U44] and the rapid expansion patterns observed thus far, it is quite possible that this number will be reached within the next few years. In any case, according to the previous analysis, the following approximations can also be made:

$$N = O(10^5)$$

$$M = O(10^1)$$

$$A = O(10^4)$$

$$P = O(10^1)$$

Thus, the BGP-4 routing table should have $O(10^6)$ entries, as indeed it does, since the exact number is currently about 0.67 * 10^6 (Table 7.1).

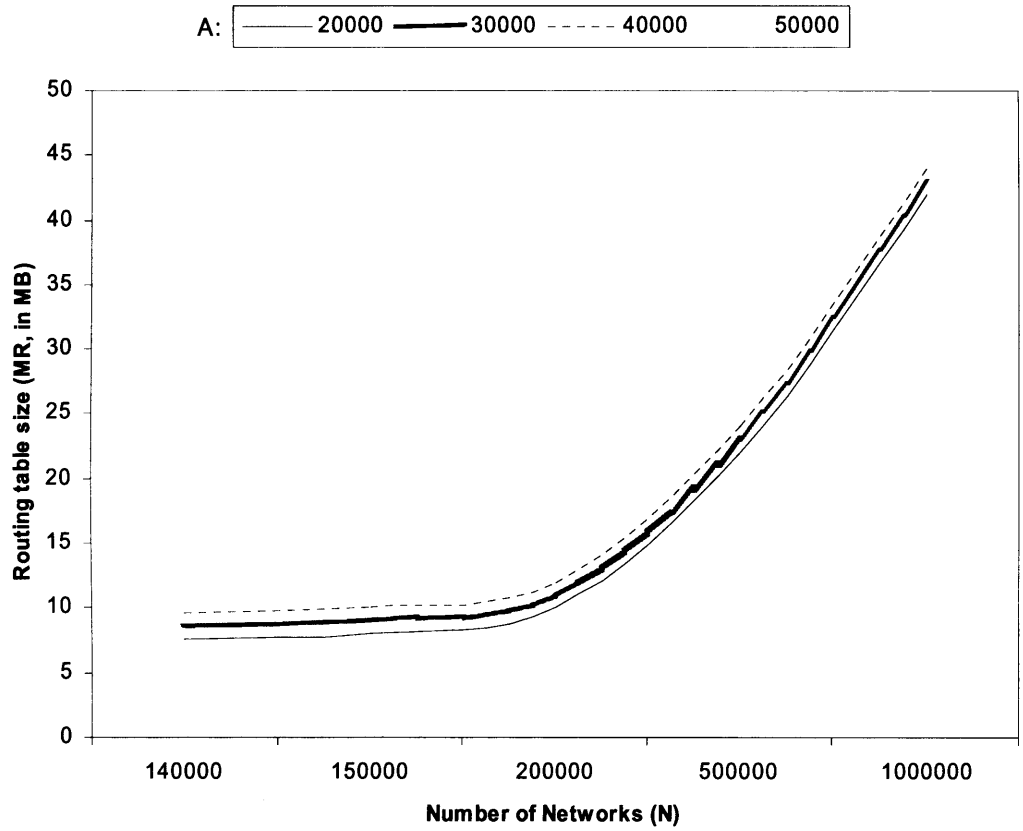


Figure 7.8: Evolution of the BGP-4 routing table size (P=10)

TRIP (actually, E-TRIP) routing table sizes can be extrapolated from the above figures via reasonable approximations, since no worldwide IP Telephony network exists to provide exact values for *N*, *M*, *A* and *P*. For a start, *N* in the case of TRIP will denote the number of IPTNs, as identified by telephone number prefixes, which, according to current international numbering plans, are over N_1 = 500,000 [U20], i.e. close to $O(10^6)$. Secondly, *A* will correspond to the number of ITADs, which is in the order of $O(10^5)$, as seen in Table 7.3. If the values of *M* and *P* above are assumed to remain the same for TRIP, then equation (11) yields a routing table size of E = $O(10^7)$; this is consistent with the $O(10^6)$ number of gateways estimated earlier on, times a reasonable $O(10^1)$ potential attributes per IP Telephony gateway (since each TRIP routing table entry essentially points to a specific gateway address and attribute combination). If $O(10^1)$ bytes are needed to store a route, such a table would require about $O(10^8)$ bytes in memory (i.e., ten times more than what Figure 7.8 indicates) and indeed there are scenarios for such consumption even in current BGP-4 networks [264].

From the close relationship of E-BGP with E-TRIP and the fact that HIT uses the latter for its route advertisement process, an estimation of the HIT routing table size can be reached. For this purpose, equation (9) can be adapted to the IP Telephony figures that have been calculated earlier on, by modifying *N*, *M*, *A* and *P* accordingly. As there are infinite possible combinations to consider and no real data to construct an explicit model, the analysis that follows focuses on the simplest case, where all changes are evenly amortised across the hierarchy and thus average values can be credibly considered for the corresponding estimations.

Clearly, the number of prefixes (N_i) advertised at each layer i ($1 \le i \le n$) will depend on the amount of aggregation achieved, compared to the immediately lower layer. Assuming an equal rate of aggregation, *r*, it can be expected that, for a hierarchy of height *n*, *N* will scale down as

$$N_i = \frac{N_1}{r^{i-1}} \quad (12)$$

where $N_1 = O(10^6)$, as already discussed. The mean distance, M_i, is a small value (less than 10, as seen above) that will be reduced slowly towards 1 as the height of the hierarchy increases, thus it can be considered a constant for the purposes of the approximation, i.e.

$$M_i \approx \mu < 10 \quad (13)$$

The average number of ITADs, A_i, at each level i of the hierarchy, can be derived from the quotient of the number of call routers, S_i, at that level, over the number of call routers per ITAD, C_i. According to equation (6) and Table 7.3,

$$A_i = \frac{S_i}{C_i} = \frac{am^{n-i}}{C_i} = O(m^{n-i}) \quad (14)$$

since both a and C_i are estimated to be in the order of $O(10^2)$. Finally, *P* can be assumed to retain the same order of magnitude across the hierarchy, i.e.

$$P = O(10^1) \quad (15)$$

Combining equations (11)-(15), produces

$$E_i = O((N_i + (M_i * A_i) * P_i) \Rightarrow$$

$$E_i = O((\frac{N_1}{r^{i-1}} + \mu * m^{n-i}) * P_i) \Rightarrow$$

$$E_i = O((\frac{10^6}{r^{i-1}} + \mu * m^{n-i}) * 10) \quad (16)$$

Assuming a hierarchy constructed using a scaling factor m = 10 and r = 10/k, equation (16) becomes

$$E_i = O((\frac{10^6}{(10/k)^{i-1}} + \mu * 10^{n-i}) * 10) \Rightarrow$$

$$E_i = O(k^{i-1} 10^{8-i} + \mu * 10^{(n+1)-i}) \quad (17)$$

If $(n+1) \leq 8$, then $1 \leq i \leq n \leq 7$ and equation (17) finally becomes

$$E_i = O(k^{i-1} 10^{8-i}) \quad (18)$$

The actual size of the routing table, in bytes, can be approximated, as above, by

$$MR_i = E_i * O(10^1) \Rightarrow$$

$$MR_i = O(k^{i-1} 10^{9-i}) \quad (19)$$

Equation (19) is plotted in Figure 7.9, which shows the decrease of routing table size for average aggregation levels of r=10, r=5 and r=2 (i.e., for k=1, k=2 and k=5, respectively). Table 7.5 summarises the preceding discussion.

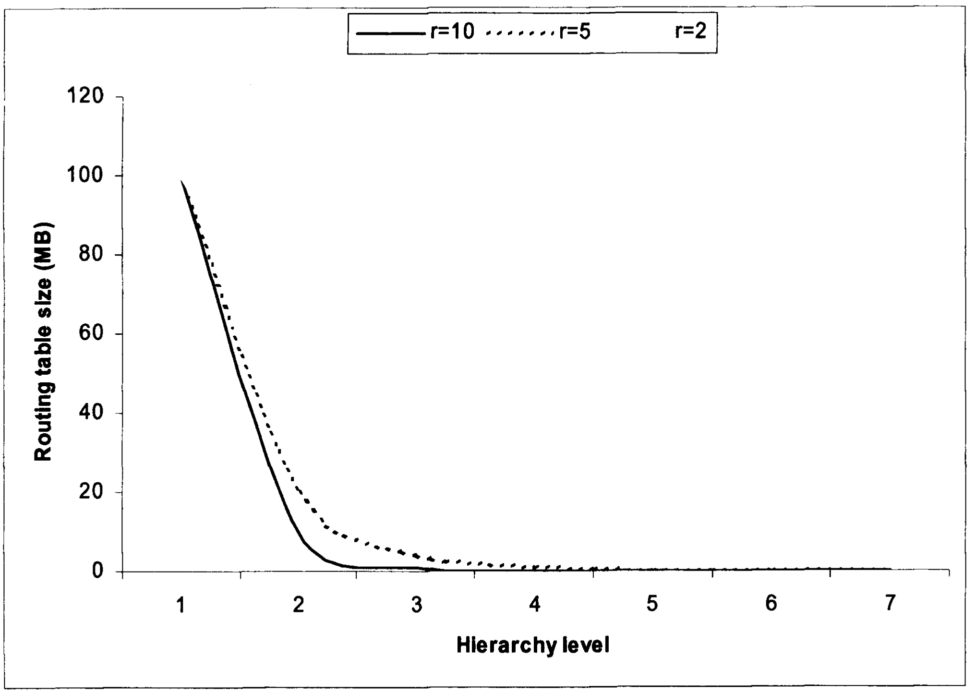


Figure 7.9: HIT routing table reduction, for various levels of aggregation

For "deeper" hierarchies (i.e., those where n > 7), Equation (16) can be simplified to

$$E_i = O(m^{n-i}) \quad (20)$$

which indicates a worst case, for which HIT becomes non-efficient for the lower layers. The above results can be generalised for other values of *m* and *r*, since, as can be seen from equations (18) and (20), the orders of magnitude remain the same.

Routing Table	**Entries**
BGP-4	$O(10^6)$
TRIP	$O(10^7)$
HIT	$O(k^{i-1} * 10^{8-i})$

Table 7.5: Comparison of BGP-4, TRIP and HIT routing tables

A number of additional remarks can be made with respect to HIT. First of all, even smaller routing table sizes are possible, since the calculations from which Figure 7.9 has resulted refer to the maximum number of call routers per hierarchy level, which is neither a requirement of HIT, nor a practical necessity. Secondly, as happens with other hierarchical protocols, a large number of levels is theoretically possible, but in practice less than 10 are adequate for covering the entire ISTN of $O(10^{10})$ terminals or more; this fact is demonstrated in practice by the GSTN and, at any rate, in current IP networks the number of levels cannot exceed 3 or 4, due to the limitations of the IPv4 address space. Third, because the hierarchy is formed via administrative action, the values of *r* and *m* used above can be perceived as general design guidelines. Fourth, because of the policy-based operation, the complexity of the protocol, as it happens with BGP-4 and TRIP, is exacerbated, resulting in scenarios that can lead to divergence [94], [135]. Fifth, the potential inefficiencies in route calculation, inherent in all hierarchical architectures, are expected to be dampened significantly by the exclusive use of Call Routers above level 0 and the small total number of levels (less than 10, as already discussed); Sixth, the exponential reduction of ITAD sizes as the height of the hierarchy increases is anticipated to maintain aggregate control traffic (e.g., routing table updates) at low levels; Seventh, a number of extensions in the areas of address aggregation, route selection and security are necessary before HITRA can be considered for commercial implementations.

In addition to a significant increase in scalability through the employment of a standardised hierarchical architecture and an exponential reduction of memory requirements, HIT greatly simplifies the call routing problem by arranging only CRs in a hierarchy, thus bypassing the need for link aggregation or physical topology changes and speeding up convergence time, compared to more complex hierarchical protocols like PNNI [11]. In addition, the source routing model followed allows the incorporation of more than one LIFT gateways in the end-to-end path (if such a need exists) and can improve Quality of Service provision through increased path stability and more fine-tuned ITAD sizes, which enable the establishment of cross-provider SLAs. Furthermore, the confederation model followed, facilitates policy enforcement among providers, a crucial property for policy-based protocols like BGP-4, TRIP and HIT.

7.6 Conclusion

Call routing is an essential process in all communication networks, voice and data. Numerous architectures and protocols have been suggested for both cases, with the hierarchical ones excelling in scalability, at the cost of added complexity and reduced accuracy in the route decision process. For IP Telephony call routing, in particular, the proposed IETF standard, TRIP, is largely based on the prevailing IP inter-domain routing protocol, BGP-4, and thus it inherits its advantages and deficiencies, especially in terms of scalability, due to the nearly flat architecture and the similar operational model used.

This chapter has described a different approach, HITRA, based on the LIFT-compliant Hierarchical IP Telephony (HIT) protocol, a generic call routing mechanism that uses a hierarchical network organisation and source routing to circumvent the anticipated scalability problems that will accompany the advent of the ISTN. HITRA reuses field-proven BGP-4 and TRIP functionality, which it extends by adding confederations and thus improves scalability in terms of convergence time and routing table size, by requiring that only location servers participate in hierarchy formation and applying other HIT-specific techniques. This way, the third component of the architecture defined in Chapter 4, call routing, is addressed.

CHAPTER 8

Conclusions

and Future Research

8.1 Overview

The advantages of digital technology, the efficiency of packet switching and the dominance of the Internet Protocol since the 1990s, have made Voice over IP an attractive, if not superior alternative to conventional telephony. However, despite intensive research and development efforts in both the academic and the commercial world, particularly during the past decade, there is a considerable amount of progress still to be made before a universal IP Switched Telephone Network becomes a reality.

This chapter summarises the main findings of the work detailed in Chapters 1-7, lists a number of areas where this work could be expanded and identifies directions that IP Telephony, in general, may or should follow in the years to come.

8.2 Main Findings and Contributions

Packet voice excels compared to conventional, circuit-switched telephony, due to its *efficiency* (stemming from the integration with data in a common, easy to administer digital backbone), potential *cost savings* (both on call charges and network operation), and the perspective for advanced *new services* (such as multimedia conferencing) to be offered to the end user. Although the economics and features of the GSTN have significantly improved during the past decades as a result of deregulation (increased market competition), digitalisation and sheer technical progress, it is reasonable to anticipate that, sooner rather than later, packet-based telephony will become commonplace, starting from the backbone. The Internet Protocol has already emerged as the preferred technology for implementing this transition, because of its universal presence in data communications (including the World Wide Web) and despite its shortcomings in terms of performance, reliability and Quality of Service.

The deployment of large-scale VoIP networks, and even more the development of a single, global IP Switched Telephone Network (ISTN), encompassing and replacing the GSTN, continues to be a complex engineering task, as indicated by the current existence of several isolated IPTNs, operated by different providers and offering limited connectivity that has to be complemented with circuit switched telephony. Solutions to the main technological challenges for overcoming this situation already exist, but they are either partial or incomplete and thus additional parameters, including the social one (i.e., technology adoption factors), will have to be taken into account before the ISTN becomes a reality. Most probably, the well established usage model of the GSTN (in terms of ease-of-use, performance and customer service) will have to be followed, with IP-specific advances in all three core areas of communication: the *LAN*, the *MAN* and the *WAN*.

The above thoughts resonate strongly with the theoretical context and the hypothesis set in Chapter 1, and they are also backed by the findings of the extensive review of packet voice technologies, including VoIP, conducted in Chapters 2 and 3 of this thesis. A new, generic IP Telephony architecture, **LIFT**, was developed as a result of these findings and, as described in Chapter 4, it tackles the problem of developing the ISTN in a unified and scalable way, by exploiting the advantages of IP while, crucially, not neglecting the benefits of the GSTN. Importantly, the deployment of the ISTN is not envisaged to happen by replacing current circuit and packet switched voice infrastructures; instead, LIFT encompasses existing technologies and, with appropriate enhancements at the LAN (via the IPTUA), the MAN (via the gateway) and the WAN (via a broader call routing strategy), it allows them to work together towards the establishment of universal VoIP connectivity.

Following the LIFT implementation guidelines of *standardisation*, *openness* and *portability*, three research prototypes, representative of the three main areas identified in the hypothesis, have been developed and evaluated. Starting with LANs, the **Configurable Audio Tool (CAT)** is a complete signalling and media client that allows users to seamlessly participate in VoIP conferences; it also proposes a number of innovative parameterisation options for similar applications, according to a component-based and object-oriented architecture, and can easily be integrated in large-scale IPTNs, as analysed in Chapter 5.

At the MAN, the novel **Voice over IP and ATM gateway (VIA)** is a modular and scalable device that achieves transparent GSTN-IP interoperability for multiple simultaneous calls. ATM was selected as a reasonable common case for such interoperability, due to its use as a carrier not only for native voice traffic, but also for VoIP (through IP over ATM) and VoATM, which, in turn, can be sourced directly from many GSTN PBXs and ATM switches. Two versions of the gateway (a TIPHON and a LIFT-based one) were developed and evaluated in Chapter 6, demonstrating the efficiency of each of the corresponding architectural frameworks and exploiting the performance of the three alternative media engines used. Although the performance of the VIA was measured to be an order of magnitude less than similar commercial products, a closer examination of the TIPHON version, which follows the same architecture as these products, reveals that, with proper amounts of optimisation and development resources, the LIFT-based version can also scale up to comparable levels.

Finally, at the WAN, the **Hierarchical IP Telephony (HIT)** protocol represents a new approach to VoIP call routing, by expanding the current (and under-researched) standard, TRIP, for scalability purposes. As the comparative evaluation of TRIP and HIT in Chapter 7 shows, a hierarchical organisation of the gateway address space stored in the call routers (location servers), presents clear benefits in terms of reduced routing table sizes, an achievement which is crucial for the successful deployment of large-scale networks. Furthermore, such architecture allows the implementation of source routing mechanisms that can improve call setup times and also allow for more complex signalling and media path configurations, on which more than one gateways can be involved end-to-end.

8.3 Extensions and Amplifications

The work presented in Chapters 4-7 can be expanded in a number of ways or directions, respectively, as follows:

- **Chapter 4**: LIFT is intentionally generic, so the extent of potential expansion is limited by design. However, more work is needed in the areas of network architecture, addressing, architecture, multicasting, security and billing. For example, the implications of using IP addresses for all terminals (both GSTN and IP) need to be further investigated, as such an approach leads to a potential exhaustion of the corresponding address space, despite techniques like CIDR [119] and NAT [311], [312], [319], thus intensifying the pressure for a transition to IPv6. Furthermore, the results of the evaluation of HIT indicate that, instead of the current, generic network organisation guidelines, a hierarchical model of operation could be specified for LIFT, and that would inevitably affect the path calculation algorithm. In addition, the path selection parameters proposed need further understanding, as these will depend on the set of gateway features, which is all but finalised yet, even for the TRIP framework [316].
- **Chapter 5**: CAT, the implementation of the audio tool architecture presented herein, will need improvements at least in the areas of the GUI, security (particularly for masquerading and denial of service threats), error concealment, signalling (H.323, H.248/MEGACO), CODEC support and playout buffer management. This list can be prolonged even more, if a feature set comparable to that of RAT, which can be considered a complete implementation in most aspects, is to be followed. It would also be interesting to assess

how CAT behaves in a VVoIP environment and, more specifically, the modifications needed in the Media Agent and the Control Agent in order for the application to be fully integrated into a conferencing tool, or even natively accommodate video traffic.

- **Chapter 6**: The VIA is vulnerable to a number of security issues, as already discussed. Beyond these, there is a large amount of features that can be added, ranging from expanded signalling and media support, to mixing and conferencing capabilities. A particular challenge, for instance, would be to replace the current PDC mechanism used for the control of the Media Gateway with an MGCP or H.248/MEGACO implementation, and add similar functionality to the CAT Call Agent, in order to use CAT as an MG. The most important enhancement to the VIA, though, would be the optimisation of its performance to commercial standards, which will require much more sophisticated hardware, including multiple physical media ports.
- **Chapter 7**: HITRA is arguably the most challenging of the three implementations presented in this thesis, given the current scarcity of research in the area of IP Telephony call routing and the complexity of the subject. An obvious enhancement is the finalisation of extensions to TRIP, or the creation of an entirely new protocol specification, down to the packet format. A mechanism for automated hierarchy formation like the one found in PNNI [11] (but more policy-driven, to accommodate provider SLAs), possibly combined with (or derived from) an established network topology generation algorithm, would also constitute a significant upgrade; similarly, a modification of the Routing Policy Specification Language (RPSL) [2] to support IP Telephony could be attractive. Route and attribute aggregation is another area where further investigation is necessary, since the scalability of both protocols (TRIP and HIT) clearly depends on it. In addition, a protocol for implementing the communication between the Signalling Server and the LS, which is now assumed to happen directly, as a result of collocation, could be interesting. Extensive large scale simulations should also be conducted, to accompany the evaluation results with empirical data, if necessary.

It should be noted that the above list is not an exhaustive but rather a restricted and prioritised one, according to the amount of implementation already completed; in other words, the suggested extensions should be straightforward, based on the work presented in this thesis. Beyond that, there is a large number of potential enhancements, driven by emerging developments in the wider IP Telephony field.

8.4 Future Directions in IP Telephony

The following areas are currently under-researched or rapidly expanding in significance, and can be expected to gain increased importance in the near future:

- *Human-Computer Interaction (HCI).* Beyond technical challenges, the VoIP usage model must approximate (if not better) the simplicity, uniformity and efficiency of the GSTN, as technology adoption research implies [276], [309]. Thus, standards should be developed for more user-friendly applications (including improvements on the GUI front, where there is still a wide variation in the "look and feel" of softphones and IP Phones) [49], [145], [307], [325], and further advances in simplifying computer use (e.g., the creation of reliable, instant-on terminals) must be achieved.
- *Security.* IP Telephony networks tend to inherit security risks from both communication worlds and the full range of threats against them is far from understood yet, as discussed in Chapter 3 [20], [249], [343], [U61], [U62]. Furthermore, there is an inevitable trade-off with usability, as increased security usually implies increased complexity and, hence, decreased levels of end user comfort with (or even rejection of) the technologies offered [32], [33]. The main application layer protocols for VoIP signalling (H.323, SIP) and traffic (RTP) are accompanied by "secure" versions or provisions, which, however, are of limited functionality, have not been widely tested and do not address potentially significant threats like VoIP spam (SPIT) [305]. TCP/IP network and transport layer security mechanisms like IPSec and TLS, combined with application layer technologies such as firewalls, seem more promising for this purpose, but their impact on delay and QoS-sensitive IP Telephony networks is still unknown and may turn out to be prohibitively high, so new protocols (or optimised versions of the existing ones) will probably be needed; this requirement is expected to be progressively accentuated, as IP Telephony expands from the backbone to the customer premises in a worldwide scale. Propelled by the need for more effective law enforcement that emerged at the start of the 21st century, surveillance (particularly lawful interception) will also become increasingly important in the years to come [105], [327]. Support for emergency services can be expected to become compulsory, too (as already mandated by the FCC for "Enhanced 911" in the United States [U10]), thereby fuelling further research efforts in this area [51], [52].

- *Regulation.* In many aspects, IP Telephony can be classified as a telecommunications service and, as such, it has long been at the centre of an ongoing debate about its regulation. In the mid-1990s, when the first publicly available VoIP applications appeared, telcos in the US and elsewhere tried, unsuccessfully, to convince their governments to ban the service, in fear of vast revenue losses [79]; at the start of the 21st century, the ability to block VoIP traffic is still in the technical specification of many corporate networks, as discussed in Chapter 3 [71], [U31], and whether to classify the service as "regulated" or "unregulated" remains highly debatable [343]. Pricing is another area where intensive research and prevailing commercial conditions will somehow have to be combined [82].
- *Call Routing.* The call routing problem in IP Telephony will take much more effort until a robust and complete standard (down to packet formats and the finite state machine level) arrives. As discussed in Chapter 7, TRIP has not gained the popularity that could be anticipated by its monopoly, and this can be attributed to the (still relatively limited) coverage of VoIP services, as well as to the preference of providers to static configuration via SLAs that allow better control and, crucially, faster times to the market; at the same time, HIT needs to mature and be tested further, before being considered for a standardisation process. Following the core argument of this thesis, the move towards the ISTN will require global connectivity, including full interoperability with the GSTN, hence automated gateway location will, inevitably, become necessary.
- *Wireless VoIP.* It is already known that Third Generation (3G) and Fourth Generation (4G) mobile voice networks will be based on IP (SIP and RTP in particular) [294], [390], meaning that wireless VoIP will progressively acquire a huge user base, which already counts nearly 2 billion subscribers [170]. In addition, wireless data networks complying with the IEEE 802.15.x (Bluetooth/WPAN) [81], 802.11 (WiFi/WLAN) [122] and 802.16 (WiMAX/WMAN) [280] standards are also expanding their coverage rapidly. Therefore, wireless VoIP (particularly mobile and VoWLAN) will continue to be one of the areas to watch for future research and development.
- *Voice and Video over IP (VVoIP).* The enhancement of VoIP services with video is a reasonable next step in the evolution of IP Telephony networks, assuming that network services like multicasting and resources, both at the edges and in the backbone, will become adequately available [160]. As detailed in Chapters 2 and 3, extensive research has been conducted in the area, particularly since the 1990s, and several standards for encoding and

transmission of voice and video already exist, but VVoIP as an integrated concept will require significant additional effort before it materialises [69], [245].

- *Application Layer Mechanisms.* Given the stability of lower layers in the TCP/IP protocol stack, the application layer is the main focus for implementing VoIP functionality and, thus, innovation. Of increasing interest are Peer-to-Peer (P2P) architectures, because of their redundancy and, with respect to this thesis, scalability [246]. The P2P area has been highlighted by the success of Skype, which, as already discussed, currently enjoys a user base of over 100 million [U48]. The Skype network model, with its partially proprietary signalling and media mechanisms [21], can sustain wireless VoIP and video, too, thus evolving into a VVoIP platform for developing new services that merit special attention, as discussed more specifically in Chapters 3 and 4.
- *Transport Layer Mechanisms.* While TCP and UDP, standardised in the early 1980s, are the long-established transport layer protocols for IP networks and, as such, are being used for VoIP, none of the two has been designed for this purpose, thus leading to workarounds for reliability (e.g., add-on retransmission techniques for UDP-based signalling), interactivity (e.g., using TCP only for media streaming) or security purposes. Clearly, a single solution to these shortcomings could be better, and maybe SCTP [355], with its improved scalability, security and ATM-style adaptation mechanisms, will emerge as such. In any case, the transition period will not be easy or brief, given the universal presence of both TCP and UDP in current IP networks, pointing to a particularly tedious upgrade process.
- *Network Layer Mechanisms.* Similarly to the transport layer, the network layer of the TCP/IP protocol stack has been monopolised by a single standard, version 4 of the Internet Protocol. The improvements of the next version are not expected to become vital in the near future, however the transition to IPv6 is considered inevitable. IP Telephony will, of course, diversely be affected [269], so extensive research will be needed on this area, too. Among other things, the issue of addressing will have to be resolved, at least to the extent of whether telephone numbers will be used as the global mechanism for this purpose (e.g., using the BCD mechanism described in Chapter 7, or a special area code for VoIP, such as the proposed 7xx in the U.S.), a feature which is obviously required for compatibility with the current GSTN usage model.
- *Quality of Service.* The shortcomings of connectionless, best-effort services like IP, in offering adequately high User and Network QoS, are well known and, as noted in Chapter 3, the research for a single, ideal solution is far from over. It cannot be predicted, at this

point, whether the answer to the overall QoS problem will be a particular (and universally deployed) technology, or a set of heuristics (e.g., broadband connectivity) which, combined, will provide the desired QoS at the network edge in conjunction with a high-performance backbone. It is likely, however, that users will be offered gradual levels of QoS, which will then somehow be reflected on their communication service charges. To be sure, MPLS appears as an increasingly stronger contender (for example, it is the solution employed at the BT 21CN network [U2]). Measuring QoS and network performance, in general, is also an area of intense ongoing research [U17], [U21].

- *New Services*. The increased intelligence of IP networks facilitates the development and deployment of innovative communication services that will drive IP Telephony further forward [295]. As discussed in Chapter 3, this refers to a variety of technological areas with multimedia as their common theme, and encourages further exploration. From simple tools like Skype, to advanced applications like streaming media (e.g., IPTV), video on demand and triple play or quad play, the interest seems to be focusing on large consumer audiences, for which data delivery (in a unicast or multicast fashion), resource efficiency and security will be areas of increased research effort in the near future [69], [286], [356], [U48].

The above list is not, of course, exhaustive, but it does indicate potential directions for immediate expansion of the work presented in Chapters 4, 5, 6 and 7, particularly in the areas of HCI, security and call routing.

8.5 Epilogue

The Internet Protocol has already established itself as the networking technology of choice for a variety of multimedia communication services, either interactive or on demand. IP Telephony is the most popular of these services, having seen rapid expansion in its market share (compared to the GSTN) over the past decade. Despite all this progress, however, a global IP Switched Telephony Network, the ISTN, has still not materialised and its advent will depend on factors that extend beyond the technical dimension of the LAN, the MAN and the WAN, well into the social field. It is the belief of the author that the philosophy, the results and, even more, the vision of this work will contribute significantly towards this objective.

REFERENCES

A list of the main references related to this thesis, compiled out of a collection of more than 1,000 in total, is included in the following two sections. *Bibliography* contains the titles of printed publications (books or papers), which are either classic in the field, unique, or very recent (i.e., published in 2004, 2005 and 2006); *Online Resources* are URLs to websites or technical standards that can be found only on the World Wide Web.

R1. Bibliography

[1] R. Akester, D. Terzis, J. Crowcroft, V. Hardman, and S. Hailes, "Java Performance for Multimedia Applications", ACM/IFIP Europar '98 Conference, Southampton, UK, September 1998

[2] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, June 1999

[3] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden, "Internet Low Bit Rate Codec (iLBC)", RFC 3951, December 2004

[4] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification", RFC 3036, January 2001

[5] F. Andreasen, and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 3435, January 2003

[6] N. Anerousis, R. Gopalakrishnan, C.R. Kalmanek, A.E. Kaplan, W.T. Marshall, P.P. Mishra, P.Z. Onufryk, K.K. Ramakrishanan, and C.J. Sreenan, "TOPS: An Architecture for Telephony over Packet Networks", IEEE Journal on Selected Areas of Communications, Vol. 17, No. 1, pp. 91-108, January 1999

[7] C.M. Aras, J.F. Kurose, D.S Reeves, and H. Schulzrinne, "Real-Time Communication in Packet-Switched Networks", Proceedings of the IEEE, Vol. 82, No. 1, pp. 122-139, January 1994

[8] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka, "Security Mechanism Agreement for the Session Initiation Protocol (SIP)", RFC 3329, January 2003

[9] G.R. Ash, "Dynamic Routing in Telecommunications Networks", McGraw-Hill, March 1998

[10] ATM Forum, "UNI 3.1 - User Network Interface Specification version 3.1", Technical Specification af-uni-0010.002, September 1994

[11] ATM Forum, "Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)", af-pnni-0055.000, March 1996

[12] ATM Forum, "ATM trunking using AAL1 for narrow band services v1.0", Technical Specification af-vtoa-0089.000, July 1997

[13] ATM Forum, "Java ATM API", Working Document, Baseline Text, BTD-SAA-API-JAVA-00.04, February 2000

[14] Y.A. Au, and R.J. Kauffirtan, "Information Technology Investment and Adoption: A Rational Expectations Perspective", Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS '03), 10 pp., Hawaii, January 2003

[15] J. Aweya, "Trunking of TDM and Narrowband Services over IP Networks", International Journal of Network Management, Vol. 13, No. 1, pp. 33-60, January/February 2003

[16] F. Baker, B. Foster, and C. Sharp, "Cisco Architecture for Lawful Intercept in IP Networks", RFC 3924, October 2004

[17] M. Bangalore, R. Kumar, H. Salama, J. Rosenberg, and Shah, "A Telephony Gateway REgistration Protocol (TGREP)", Internet Draft, draft-ietf-iptel-tgrep-06, July 2005

[18] D. Bansal, J.Q. Bao, W.C. Lee, "QoS-Enabled Residential Gateway Architecture", IEEE Communications Magazine, Vol. 41, No. 4, pp. 83-89, April 2003

[19] G. Barberis, "Buffer Sizing of a Packet Voice Receiver", IEEE Transactions on Communications, Vol. COM-29, No. 2, pp. 152-156, February 1981

[20] R. Barbieri, D. Bruschi, and E. Rosti, "Voice over IPsec: Analysis and Solutions", Proceedings of the IEEE 18th Annual Computer Security Applications Conference (ACSAC '02), pp. 261-270, December 2002

[21] S. Baset, and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", Technical Report CUCS-039-04, Computer Science Department, Columbia University, New York, NY, September 2004

[22] T. Bates, R. Chandra, and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP", RFC 2796, April 2000

[23] T. Bates, Y. Rekhter, R. Chandra, and D. Katz, "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000

[24] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004

[25] M. Bearden, L. Denby, B. Karacali, J. Meloche, and D.T. Stott, “Assessing Network Readiness for IP Telephony”, Proceedings of the IEEE International Conference on Communications (ICC ’02), pp. 2568-2572, New York, NY, April-May 2002

[26] J.C. Bellamy, “Digital Telephony”, Wiley, March 2000

[27] S. Bellovin, J. Schiller, and C. Kaufman (eds.), “Security Mechanisms for the Internet”, RFC 3631, December 2003

[28] H.C. Berkowitz, “Designing Addressing Architectures for Routing and Switching”, Macmillan Technical Publishing, September 1998

[29] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, “A Framework for Integrated Services Operation over Diffserv Networks ”, RFC 2998, November 2000

[30] R.K. Bhattacharyya, “New Challenges for Telephone Companies to Secure Switching Systems”, Proceedings of the IEEE 25th Annual International Carnahan Conference on Security Technology, 1991, pp. 41-46, Taipei, Taiwan, October 1991

[31] T. Bially, B. Gold, and S. Seneff, “A Technique for Adaptive Voice Flow Control in Integrated Packet Networks”, IEEE Transactions on Communications, Vol. COM-28, No. 3, pp. 325-333, March 1980

[32] H. Bidgoli (ed.), “Handbook of Information Security”, Wiley, January 2006

[33] M. Bishop, “Introduction to Computer Security”, Addison-Wesley, November 2004

[34] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services”, RFC 2475, December 1998

[35] G. Bochman, and P. Mondain-Monval, “Design Principles for Communication Gateways”, IEEE Journal on Selected Areas in Communications, Vol. 8, No. 1, pp. 12-21, January 1990

[36] J.-C. Bollot, “Characterizing End-to-End Packet Delay and Loss in the Internet”, Journal of High-Speed Networks, Vol. 2, No. 3, pp. 305-323, December 1993

[37] J.-C. Bollot, and A. Vega-Garcia, “Control Mechanisms for Packet Audio in the Internet”, Proceedings of IEEE INFOCOM ’96, Vol. 1, pp. 232-239, March 1996

[38] J.-C. Bollot, and S. Fosse-Parisis, “Adding Voice to distributed games on the Internet”, Proceedings of IEEE INFOCOM ’98, Vol. 2, pp. 480-487, March-April 1998

[39] M. Borella, D. Grabelsky, J. Lo, and K. Taniguchi, “Realm Specific IP (RSIP): Protocol Implementation”, RFC 3103, October 2001

[40] M. Borella, J. Lo, D. Grabelsky, and G. Montenegro, “Realm Specific IP (RSIP):

Framework", RFC 3102, October 2001

[41] C.M. Bowman, P. Danzig, D. Hardy, U. Manber, and M. Schwartz, "The Harvest Information Discovery and Access System", Computer Networks and ISDN Systems, Vol. 28, No. 1-2, pp. 119-126, December 1995

[42] R. Braden, D. Clark, and S. Shenker, "Integrated -Services in the Internet Architecture: an Overview", RFC 1633, June 1994

[43] R. Braden (ed.), L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997

[44] P.T. Brady, "A Technique for Investigating On-Off Patterns of Speech", Bell System Technical Journal, Vol. 44, No. 1, pp. 1-22, January 1965

[45] S. Bryant (ed.), and P. Pate (ed.), "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005

[46] T. Bu, L. Gao, and D. Towsley, "On Characterizing BGP Routing Table Growth", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02), Vol. 3, pp. 2185-2189, November 2002

[47] P.M. Bull, P.R. Canyon, and P.R. Limb, "Residential Gateways", BT Technology Journal, Vol. 20, No. 2, pp. 73-81, April 2002

[48] R. Caceres, N. Duffield, and T. Friedman, "Impromptu measurement infrastructures using RTP", Proceedings of IEEE INFOCOM '02, Vol. 3, pp. 1490-1499, June 2002

[49] J.J. Cadiz, A. Narin, G. Jancke, A. Gupta, and M. Boyle, "Exploring PC-Telephone Convergence with the Enhanced Telephony Prototype", Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '04), pp. 215-222, Vienna, Austria, April 2004

[50] M.C. Caesar, S. Balaraman, and D. Ghosal, "A Comparative Study of Pricing Strategies for IP Telephony", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '00), pp. 344-349, November 2000

[51] K. Calberg, and R. Atkinson, "General Requirements for Emergency Telecommunication Service (ETS)", RFC 3690, February 2004

[52] K. Calberg, and R. Atkinson, "IP Telephony Requirements for Emergency Telecommunication Service (ETS)", RFC 3690, February 2004

[53] G. Camarillo, "SIP Demystified", McGraw-Hill, April 2001

[54] G. Camarillo, "The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds", Wiley, December 2005

[55] S.J. Campanella, "A survey of speech bandwidth compression techniques", IRE Transactions on Audio, Vol. 6, No. 5, pp. 104-116, September 1958

[56] B. Campbell (ed.), J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002

[57] G. Carle and E.W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications", IEEE Network, Vol. 11, No. 6, pp. 24-36, November/December 1997

[58] J.D. Case, J. Davin, M. Fedor, and M.L. Schoffstall, "Introduction to the Simple Gateway Monitoring Protocol", RFC 1028, November 1987

[59] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990

[60] S. Casner, and S. Deering, "First IETF Internet audiocast", Computer Communications Review, Vol. 22, No. 3, July 1992

[61] F.C. Castello, R. Balbinot, J.G. Silveira, and P.M. Santos, "A Robust Architecture for IP Telephony Systems Interconnection", Proceedings of the 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '03), Vol. 2, pp. 593-596, Victoria, Canada, August 2003

[62] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, August 1996

[63] V.G. Cerf, "On the Evolution of Internet Technologies", Proceedings of the IEEE, Vol. 92, No. 9, pp. 1360-1370, September 2004

[64] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute", RFC 1997, August 1996

[65] A. Chakrabarti, and G. Manimaran, "Internet Infrastructure Security: A Taxonomy", IEEE Network, Vol. 16, No. 6, pp. 13-21, November-December 2002

[66] C.-Y. Chang, and M.-S. Chen, "On Building an Internet Gateway for Internet Telephony", Proceedings of the 2000 IEEE International Conference on Multimedia and Expo (ICME 2000), Vol. 3, pp. 1771-1774, New York, NY, July-August 2000

[67] S. Chatterjee, B. Tulu, T. Abhichandani, and H. Li, "SIP-Based Enterprise Converged Networks for Voice/Video-Over-IP: Implementation and Evaluation of Components", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 10, pp. 1921-1933, October 2005

[68] W.-E. Chen, C.Y. Su, and J.-H. Weng, "Development of IPv6-IPv4 Translation

Mechanisms for SIP-based VoIP Applications", Proceedings of the IEEE 19th International Conference on Advanced Information Networking and Applications (AINA '05), pp. 819-823, March 2005

[69] S. Cherry, "The Battle for Broadband", IEEE Spectrum, Vol. 42, No. 1, pp. 24-29, January 2005

[70] S. Cherry, "Seven Myths About Voice over IP", IEEE Spectrum, Vol. 42, No. 3, pp. 52-57, March 2005

[71] S. Cherry, "The VoIP Backlash", IEEE Spectrum, Vol. 42, No. 10, pp. 61-63, October 2005

[72] H.M. Chong, and H.S. Matthews, "Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems", Proceedings of the 2004 IEEE International Symposium on Electronics and the Environment, pp. 106-111, Phoenix, AR, May 2004

[73] W.C. Chu, "Speech Coding Algorithms: Foundation and Evolution of Standardized Coders", Wiley, May 2003

[74] D. Clark, "Modularity and Efficiency in Protocol Implementation", RFC 817, July 1982

[75] D. Clark, L. Chapin, V. Cerf, R. Braden, and R. Hobby, "Towards the Future Internet Architecture", RFC 1287, December 1991

[76] D. Cohen, "Specifications for the Network Voice Protocol (NVP)", RFC 741, November 1977

[77] R. Cole, "Experience and Analysis of Network Interconnection", IEEE Journal on Selected Areas in Communications, Vol. 8, No. 1, pp. 49-56, January 1990

[78] R.G. Cole, and J.H. Rosenbluth, "Voice over IP Performance Monitoring", Proceedings of ACM SIGCOMM, Vol. 31, No. 2, pp. 9-24, April 2001

[79] D. Collins, "Carrier Grade Voice over IP", McGraw-Hill, September 2002

[80] A. Conte, L.-P. Anquetil, and T. Levy, "Experiencing Megaco Protocol for Controlling Non-decomposable VoIP Gateways", Proceedings of the 8th IEEE International Conference on Networks (ICON 2000), pp. 105-111, Singapore, September 2000

[81] T. Cooklev, "Wireless Communication Standards: A Study of IEEE 802.11, 802.15 and 802.16", IEEE Press, October 2004

[82] C. Courcoubetis, and R. Weber, "Pricing communication Networks: Economics, Technology and Modelling", Wiley, March 2003

[83] G. J. Coviello, "Comparative discussion of circuit- vs. packet-switched voice", IEEE

Transactions on Communications, Vol. COM-27, pp.1153-1159, August 1979

[84] D. Crocker, and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997

[85] T. Dagiuklas, and P. Galiotos, "Architecture and Design for an Enhanced H.323 VoIP Gateway", Proceedings of the IEEE International Conference on Communications (ICC '02), Vol. 2, pp. 1209-1213, New York, NY, April-May 2002

[86] J. Davin, J.D. Case, M. Fedor, and M.L. Schoffstall, "A Simple Gateway Monitoring Protocol", RFC 1028, November 1987

[87] M. Day, S. Aggarwal, G. Mohr, and J. Vincent, "Instant Messaging / Presence Protocol Requirements", RFC 2779, February 2000

[88] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000

[89] J.H. de Souza Pereira, J. Guilherme, and P.F. Rosa, "Development of MGs in a Next Generation Network with MEGACO/H.248 Support", Proceedings of the 12th IEEE International Conference on Networks (ICON 2004), Vol. 1, pp. 239-243, Singapore, November 2004

[90] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998

[91] B.J. Dempsey, "A new error control scheme for packetized voice over high-speed local area networks", Proceedings of the IEEE 18th Conference on Local Computer Networks, pp. 91-100, September 1993

[92] F.T.H. den Hartog, M. Balm, C.M. de Jong, J.J.B Kwaaitaai, "Convergence of Residential Gateway Technology", IEEE Communications Magazine, Vol. 42, No. 5, pp. 138-143, May 2004

[93] T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999

[94] X. Dimitropoulos, and G.F. Riley, "Large-Scale Simulation Models of BGP", Proceedings of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems 2004 (MASCOTS 2004), pp. 287-294, Volendam, The Netherlands, October 2004

[95] S. Donovan, "The SIP INFO Method", RFC 2976, October 2000

[96] T. Dorsey, "CU-SeeMe Videoconferencing Software", ConneXions - The Interoperability Report, Vol. 9, No. 3, pp. 42-45, March 1995

[97] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997

[98] A. Durand, and C. Huitema, "The H-Density Ratio for Address Assignment Efficiency An Update on the H Ratio", RFC 3194, November 2001

[99] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000

[100] B. Duysburgh, S. Vanhastel, B.D. Vreese, C. Petrisor, and P. Demeester, "On the Influence of Best-Effort Network Conditions on the Perceived Speech Quality of VoIP Connections", Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN '01), pp. 334-339, Scottsdale, AR, CA, October 2001

[101] ECMA, "Private Integrated Services Network (PISN) - Circuit Mode Bearer Services - Inter-Exchange Signalling Procedures and Protocol", ECMA-143, December 2001

[102] M. Elkins, D. Del Torto, R. Levien, and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001

[103] H. Eriksson, "MBONE: The Multicast Backbone", Communications of the ACM, Vol. 37, No. 8, pp. 54-60, August 1994

[104] A. Estepa, R. Estepa, and J.M. Vozmediano, "On the Suitability of the E-Model to VoIP Networks", Proceedings of the 7th IEEE Symposium on Computers and Communications (ISCC 2002), pp. 511-516, July 2002

[105] ETSI, "Telecommunications Security; Lawful Interception (LI), Requirements of Law Enforcement Agencies", Specification TS 101 331 V1.1.1, August 2001

[106] ETSI, "Telecommunications and Internet Protocol Harmonisation over Networks (TIPHON) Release 4; Open Settlement Protocol (OSP) for Inter-Domain Pricing, Authorization and Usage Exchange", Specification TS 101 321 V4.1.1, November 2003

[107] ETSI, "Telecommunications and Internet Protocol Harmonisation over Networks (TIPHON) Release 4 Definition", Specification TR 101 301 V3.1.1, April 2004

[108] ETSI, "TISPAN_NGN - Release 1: Release Definition", Draft TR 00001 V0.4.2, June 2006

[109] P. Faltstrom, and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004

[110] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999

[111] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic

Engineering for IP Networks", IEEE Network, Vol. 14, No. 2, pp. 11-19, March-April 2000

[112] V. Fineberg, "A Practical Architecture for Implementing End-to-End QoS in an IP Network", IEEE Communications Magazine, Vol. 40, No. 1, pp. 122-130, January 2002

[113] J. L. Flanagan, "Speech Analysis, Synthesis and Perception", Springer-Verlag, May 1972

[114] J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, J. M. Tribolet, "Speech Coding", IEEE Transactions on Communications, Vol. COM-27, No. 4, pp. 710-737, April 1979

[115] B. Fortz, J. Rexford, and M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols", IEEE Communications Magazine, Vol. 40, No. 10, pp. 118-124, October 2002

[116] Frame Relay Forum, "Voice over Frame Relay Implementation Agreement", FRF.11.1, December 1998

[117] R. Frederick, "Experiences with real-time software video compression", Proceedings of the 6th International Workshop on Packet Video, pp. F1.1-F1.4, Portland, OR, September 1994

[118] T. Friedman (ed.), R. Caceres (ed.), and A. Clark (ed.), "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003

[119] V. Fuller, T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, September 1993

[120] J.J. Garrahan, P.A. Russo, K. Kitami, and R. Kung, "Intelligent Network Overview", IEEE Communications Magazine, Vol. 31, No. 3, pp 30-36, March 1993

[121] S. Garg, and M. Kappes, "Admission Control for VoIP Traffic in IEEE 802.11 Networks", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03), pp. 3514-3518, December 2003

[122] M. Gast, "802.11 Wireless Networks: The Definitive Guide", O'Reilly, May 2005

[123] General Services Administration, "Telecommunications: Glossary of Communication Terms", U.S. Federal Standard 1037C, August 1996

[124] N. Ghani, S. Dixit, and T.-S. Wang, "On IP over WDM: A Retrospective", IEEE Communications Magazine, Vol. 41, No. 9, pp. 42-45, September 2003

[125] J. Glassman, W. Kellerer, and H. Muller, "Service Development and Deployment in H.323 and SIP", IEEE Communication Surveys and Tutorials, Vol. 5, No. 2, pp. 2-17,

Fourth Quarter 2003

[126] R.H. Glitho, “Advanced Services Architectures for Internet Telephony: A Critical Overview”, IEEE Network, Vol. 14, No. 4, pp. 38-44, July-August 2000

[127] B. Gold, “Digital Speech Networks”, Proceedings of the IEEE, Vol. 65, No. 12, pp. 1636-1658, December 1977

[128] B. Goode, “Voice over Internet Protocol (VoIP)”, Proceedings of the IEEE, Vol. 90, No. 9, pp. 1495-1517, September 2002

[129] B. Goodman, “Internet Telephony and Modem Delay”, IEEE Network, pp. 8-16, May 1999

[130] W. Goralski, “Frame Relay for High-Speed Networks”, Wiley, February 1999

[131] M.G. Graff, and K.R. van Wyk, “Secure Coding: Principles and Practices”, O’Reilly, July 2003

[132] R. M. Gray, “The 1974 Origins of VoIP”, IEEE Signal Processing Magazine, pp. 87-90, July 2005

[133] P.E. Green, “Protocol Conversion”, IEEE Transactions on Communications, Vol. COM-34, No. 3, pp. 257-268, March 1986

[134] N. Greene, M. Ramalho, and B. Rosen, “Media Gateway Control Protocol Architecture and Requirements ”, RFC 2805, April 2000

[135] T.G. Griffin, and G. Wilfong, “An Analysis of BGP Convergence Properties”, Proceedings of ACM SIGCOMM, pp. 277-288, August 1999

[136] C. Groves, M. Pantaleo, T. Anderson, and T. Taylor (eds.), “Gateway Control Protocol Version 1”, RFC 3525, June 2003

[137] J.G. Gruber, “Delay Related Issues in Integrated Voice and Data Networks”, IEEE Transactions on Communications, Vol. COM-29, No. 6, pp. 1478-1490, June 1981

[138] J.G. Gruber, and N.H. Le, “Performance Requirements for Integrated Voice/Data Networks”, IEEE Journal on Selected Areas in Communications, Vol. 1, No. 6, pp. 981-1005, December 1983

[139] J.G. Gruber, and L. Strawczynski, “Subjective Effects of Variable Delay and Speech Clipping in Dynamically Managed Voice Systems”, IEEE Transactions on Communications, Vol. COM-33, No. 8, pp. 801-808, August 1985

[140] V. Gurbani (ed.), A. Brusilovsky, I. Faynberg, J. Gato, H. Lu, and M. Unmehopa, “The SPIRITS (Services in PSTN requesting Internet Services) Protocol”, RFC 3910, October 2004

[141] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999

[142] O. Hagsand, K. Hanson, and I. Marsh, "Measuring Internet Telephony Quality: Where Are We Today?", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '99), Vol. 3, pp. 1838-1842, December 1999

[143] O. Hagsand, K. Hanson, and I. Marsh, "Sicsophone: A Low-Delay Internet Telephony Tool", Proceedings of the 29th Euromicro Conference (EUROMICRO '03), pp. 189-195, September 2003

[144] B. Halabi, "Internet Routing Architectures", Cisco Press, August 2000

[145] R. Halstead-Nussloch, "The Design of Phone-Based Interfaces for Consumers", Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '89), pp. 347-352, Austin, TX, May 1989

[146] M. Hamdi, O. Verscheure, J.-P. Hubaux, I. Dalgic, and P. Wang, "Voice Service Interworking for PSTN and IP Networks", IEEE Communications Magazine, Vol. 37, No. 5, pp. 104-111, May 1999

[147] L-N. Hamer, B. Gage, and H. Shieh, "Framework for Session Set-up with Media Authorization", RFC 3521, April 2003

[148] D. Hampton, D. Oran, H. Salama, and D. Shah, "The IP Telephony border Gateway Protocol (TBGP)", Internet Draft, draft-ietf-iptel-glp-tbgp-01, June 1999

[149] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn, "Point-to-Point Tunneling Protocol", RFC 2637, July 1999

[150] M. Handley, and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998

[151] F. Hao, and P. Koppol, "An Internet Scale Simulation Setup for BGP", ACM SIGCOMM Computer Communication Review, Vol. 33, No. 3, pp. 43-57, July 2003

[152] M. Handley, C. Perkins, and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000

[153] V. Hardman, I. Kouvelas, M.A. Sasse, and A. Watson, "A Packet Loss Robust-Audio Tool for Use over the Mbone", Department of Computer science, University College London, Research Note RN/96/8, August 1996

[154] V. Hardman, M.A. Sasse, M. Handley, and A. Watson, "Reliable Audio for Use over the Internet", Proceedings of INET95, Hawaii, June 1995

[155] V. Hardman, M.A. Sasse, and I. Kouvelas, "Successful Multiparty Audio

Communication over the Internet", Communications of the ACM, Vol. 41, No. 5, pp. 74-80, May 1998

[156] W.C. Hardy, "VoIP Service Quality", McGraw-Hill, February 2003

[157] S. Hares, and A. Retana, "BGP-4 Implementation Report", RFC 4276, January 2006

[158] J. Hautakorpi (ed.), G. Camarillo, M. Bhatia, R. Penfield, and A. Hawrylyshen, "SIP (Session Initiation Protocol)-Unfriendly Functions in Current Communication Architectures", Internet Draft, draft-camarillo-sipping-sbc-funcs-02, September 2005

[159] O. Hersent, J.-P. Petit, and D. Gurle, "IP Telephony - Deploying Voice-over-IP Protocols", Wiley, January 2005

[160] O. Hersent, J.-P. Petit, and D. Gurle, "Beyond VoIP Protocols: Understanding Voice Technology and Networking Techniques for IP Telephony ", Wiley, January 2005

[161] T. Hirata, S. Matsui, T. Yokoyama, M. Mizutani, and M. Terada, "A High Speed Protocol Processor to Boost Gateway Performance", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '90), Vol. 3, pp. 1426-1430, December 1990

[162] J.-M. Ho, J.-C. Hu, and P. Steenkiste, "A Conference Gateway Supporting Interoperability Between SIP and H.323", Proceedings of the 9th ACM International Conference on Multimedia (MM '01), Vol. 9, pp. 421-430, Ottawa, Canada, September 2001

[163] O. Hodson, S. Varakliotis, and V. Hardman, "A Software Platform for Multiway Audio Distribution over the Internet", IEE Colloquium on Music Technology, p.p. 4/1-4/6, November 1998

[164] T.F. Houghton, E.C. Schloemer, E.S. Szurkowski, and W.P. Weber, "A Packet Telephony Gateway for Public Network Operators", Proceedings of the XVI World Telecom Congress, International Switching Symposium (ISS '97), pp. 35-44, Toronto, Canada, September 1997

[165] C. Huitema, "The H Ratio for Address Assignment Efficiency", RFC 1715, November 1994

[166] C. Huitema, J. Cameron, P. Mouchtaris, and D. Smyk, "An Architecture for Residential Telephony Service", IEEE Internet Computing, pp. 73-82, May-June 1999

[167] C. Huitema, "Routing in the Internet", Prentice Hall, November 1999

[168] G. Huston, "Analysing the Internet BGP Routing Table", Internet Protocol Journal, Vol. 4, No. 1, pp. 2-15, March 2001

[169] IAB, and IESG, "IETF Policy on Wiretapping", RFC 2804, May 2000

[170] ITU-D, "World Telecommunication/ICT Development Report 2006", March 2006

[171] ITU-R, "Method for objective measurements of perceived audio quality", Recommendation BS.1387, November 2001

[172] ITU-T, "The international public telecommunication numbering plan", Recommendation E.164, February 2005

[173] ITU-T, "Traffic Routing", Recommendation E.170, October 1992

[174] ITU-T, "Comparative metrics for network performance management", Recommendation E.437, May 1999

[175] ITU-T, "Terms and definitions of traffic engineering", Recommendation E.600, March 1993

[176] ITU-T, "Recommendation G.114 - One-way Transmission Time", Recommendation E.600, May 2003

[177] ITU-T, "The E-model, a computational model for use in transmission planning", Recommendation G.107, March 2005

[178] ITU-T, "Talker echo and its control", Recommendation G.131, November 2003

[179] ITU-T, "Echo suppressors", Recommendation G.164, November 1998

[180] ITU-T, "Digital Echo Cancellers", Recommendation G.168, March 2003

[181] ITU-T, "Pulse code modulation (PCM) of voice frequencies", Recommendation G.711, November 1988

[182] ITU-T, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s", Recommendation G.723.1, March 1996

[183] ITU-T, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)", Recommendation G.729, March 1996

[184] ITU-T, "Call Signalling Protocols and Media Stream Packetisation for Packet-Based Multimedia Communication Systems", Recommendation H.225.0, July 2003

[185] ITU-T, "H.323 security framework: Security framework for H-series (H.323 and other H.245-based) multimedia systems", Recommendation H.235.0, September 2005

[186] ITU-T, "Control protocol for Multimedia Communication", Recommendation H.245, October 2005

[187] ITU-T, "Gateway Control Protocol: Version 3", Recommendation H.248.1, September 2005

[188] ITU-T, "Packet-Based Multimedia Communications Systems", Recommendation H.323,

July 2003

[189] ITU-T, "Generic functional protocol for the support of supplementary services in H.323", Recommendation H.450.1, July 2003

[190] ITU-T, "B-ISDN ATM Adaptation Layer specification: Type 1 AAL", Recommendation I.363.1, August 1996

[191] ITU-T, "B-ISDN ATM Adaptation Layer specification: Type 2 AAL", Recommendation I.363.1, November 2000

[192] ITU-T, "B-ISDN ATM Adaptation Layer specification: Type 3/4 AAL", Recommendation I.363.1, August 1996

[193] ITU-T, "B-ISDN ATM Adaptation Layer specification: Type 5 AAL", Recommendation I.363.1, August 1996

[194] ITU-T, "Methods for subjective determination of transmission quality", Recommendation P.800, August 1996

[195] ITU-T, "Subjective performance assessment of telephone-band and wideband digital CODECs", Recommendation P.830, February 1996

[196] ITU-T, "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs", Recommendation P.861, February 1998

[197] ITU-T, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs", Recommendation P.862, February 2001

[198] ITU-T, "Introduction to CCITT Signalling System No. 7", Recommendation Q.700, March 1993

[199] ITU-T, "Functional description of the message transfer part (MTP) of Signalling System No. 7", Recommendation Q.701, March 1993

[200] ITU-T, "Signalling System No. 7 - ISDN User Part functional description", Recommendation Q.761, December 1999

[201] ITU-T, "ISDN User-Network Interface Layer 3 Specification for Basic Call Control", Recommendation Q.931, May 1998

[202] ITU-T, "Digital subscriber signalling system No. 1 - Generic procedures for the control of ISDN supplementary services", Recommendation Q.932, May 1998

[203] ITU-T, "General series Intelligent Network Recommendation structure", Recommendation Q.1200, September 1997

[204] ITU-T, "Digital Subscriber Signalling System No. 2 (DSS 2); User-Network Interface

(UNI) Layer 3 Specification for Basic Call/Connection Control", Recommendation Q.2931, February 1995

[205] ITU-T, "Procedures for real-time Group 3 facsimile communication over IP networks", Recommendation T.38, September 2005

[206] ITU-T, "Data protocols for multimedia conferencing", Recommendation T.120, July 1996

[207] ITU-T, "Abstract Syntax Notation 1 (ASN.1) - Specification of Basic Notation", Recommendation X.680, July 2002

[208] ITU-T, "General requirements for interworking with Internet protocol (IP)-based networks", Recommendation Y.1401, October 2000

[209] ITU-T, "General overview of NGN", Recommendation Y.2001, December 2004

[210] V. Jacobson, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM, Vol. 18, No. 4, pp. 158-173, August 1988

[211] N.S. Jayant and E.Y. Chen, "Audio Compression: Technology and Applications", AT&T Technical Journal, Vol. 74, No, 2, pp. 23-34, March 1995

[212] W. Jiang, K. Koguchi, and H. Schulzrinne, "QoS Evaluation of VoIP Endpoints", Proceedings of the IEEE International Conference on Communications (ICC '03), Vol. 3, pp. 1917-1921, Anchorage, AL, May 2003

[213] W. Jiang, and H. Schulzrinne, "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality", Proceedings of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '00), Chapel Hill, NC, June 2000

[214] W. Jiang, and H. Schulzrinne, "Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality Under Bursty Loss", Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '02), Miami, FL, May 2002

[215] A. Johnston, "SIP, P2P, and Internet Communications", Internet Draft, draft-johnston-sipping-p2p-ipcom-01, March 2005

[216] A. Johnston, S. Donovan, R. Sparks, C. Cunningham, and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", RFC 3665, December 2003

[217] D.W. Jordan, and P. Smith, "Mathematical Techniques", Oxford University Press, June 2002

[218] M. Karam, and F.J. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic Over the

Internet", Proceedings of IEEE INFOCOM '01, Vol. 2, pp. 824-833, April 2001

[219] K. Katrinis, G. Parissidis, and B. Plattner, "A Comparison of Frameworks for Multimedia Conferencing: SIP and H.323", Proceedings of the 8th IASTED International Conference on Internet Multimedia Systems and Applications (IMSA), August 2004

[220] R. Kaza, and S. Asadullah, "Cisco IP Telephony", Cisco Press, November 2004

[221] C.A. Kent, and J.C. Mogul, "Fragmentation considered harmful", Proceedings of ACM SIGCOMM, pp. 390-401, August 1987

[222] S. Kent, and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[223] S. Keshav, "An Engineering Approach to Computer Networking", Addison-Wesley, June 1997

[224] F. Khan, "Traffic Engineering with Multiprotocol Label Switching", Proceedings of the IEEE E-Tech 2004, pp. 61-67, Karachi, July 2004

[225] L. Kleinrock, and F. Kamoun, "Hierarchical Routing for Large Networks: Performance Evaluation and Optimization", Computer Networks, Vol. 1, No. 3, pp. 155-174, January 1977

[226] L. Kleinrock, and F. Kamoun, "Stochastic Performance Evaluation of Hierarchical Routing for Large Networks", Computer Networks, Vol. 3, No. 5, pp. 337-353, November 1977

[227] J. Klensin (ed.), "Simple Mail Transfer Protocol", RFC 2821, April 2001

[228] K. Knightson, N. Morita, and T. Towle, "NGN Architecture: Generic Principles, Functional Architecture, and Implementation", IEEE Communications Magazine, Vol. 43, No. 10, pp. 49-56, October 2005

[229] A.M. Kondoz, "Digital Speech: Coding for Low Bit Rate Communication Systems", Wiley, September 2004

[230] T. Koren, S. Casner, J. Geevarghese, B. Thompson, and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, July 2003

[231] A. Kos, B. Klepec, and S. Tomazic, "Techniques for Performance Improvement of VoIP Applications", Proceedings of the IEEE 11th Mediterranean Electrotechnical Conference (MELECON '02), pp. 250-254, Cairo, May 2002

[232] I. Kouvelas, "A Combined Network, System and User Based Approach to Improving

the Quality of Multicast Audio", PhD Thesis, Department of Computer Science, University College London, May 1998

[233] I. Kouvelas, and V. Hardman, "Overcoming Workstation Scheduling Problems in a Real-Time Audio Tool", Proceedings of the Winter USENIX Conference, pp. 235-242, Anaheim, CA, January 1997

[234] A. Kumar K., and T. Malhotra, "A Multi-Signaling Protocol Architecture for Voice over IP Terminal", Proceedings of IEEE INFOCOM '04, Vol. 2, pp. 1191-1199, March 2004

[235] V. Kumar, M. Korpi, and S. Sengodan, "IP Telephony with H.323", Wiley, May 2001

[236] C. Kunziger (ed.), "Protocol for the Exchange of Inter-Domain Routing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs", ISO/IEC 10747, April 1994

[237] J.F. Kurose, and K.W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison-Wesley, May 2004

[238] E.T. Lakay, and J.I. Agbinya, "Security Issues in SIP Signalling in Wireless Networks and Services", Proceedings of the IEEE 4th International Conference on Mobile Business (ICMB '05), pp. 639-642, Sydney, July 2005

[239] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002

[240] H.H. Lee, and C.K. Un, "A Study of On-Off Characteristics of Conversational Speech", IEEE Transactions on Communications, Vol. COM-34, No. 6, pp. 630-837, June 1986

[241] J. Lennox, H. Schulzrinne, and J. Rosenberg, "Common Gateway Interface for SIP", RFC 3050, January 2001

[242] J. Lennox, X. Wu, and H. Schulzrinne, "Call Processing Language (CPL): A Language for User Control of Internet Telephony Services", RFC 3880, October 2004

[243] O. Levin, "Telephone Number Mapping (ENUM) Service Registration for H.323", RFC 3762, April 2004

[244] B. Li, M. Hamdi, D. Iang; X.-R. Cao, Y.T. Hou, "QoS-Enabled Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges", IEEE Communications Magazine, Vol. 38, No. 4, pp. 54-61, April 2000

[245] Z.-N. Li, and M.S. Drew, "Fundamentals of Multimedia", Pearson Education, June 2004

[246] D. Liben-Nowell, H. Balakrishnan, and David Karger, "Analysis of the Evolution of Peer-to-Peer Systems", Proceedings of the 21st Annual Symposium on Principles of

Distributed Computing (PODC '02), pp. 233-242, Monterey, CA, July 2002

[247] J. Liesenborgs, W. Lamotte, and F. Van Reeth, "Voice over IP with JVOIPLIB and JRTPLIB", Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN '01), pp. 346-347, Tampa, FL, November 2001

[248] J. Light, and A. Bhuvaneshwari, "Performance Analysis of Audio Codecs Over Real-Time Transmission Protocol (RTP) for Voice Services Over Internet Protocol", Proceedings of the 2nd Annual Conference on Communication Networks and Services Research, pp.351-356, May 2004

[249] J. Loughney, M. Tuexen, Ed., and J. Pastor-Balbas, "Security Considerations for Signalling Transport (SIGTRAN) Protocols", RFC 3788, June 2004

[250] C. Low, "The Internet Telephony Red Herring", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '96), pp. 72-80, November 1996

[251] J. Luciani, G. Armitage, J. Halpern, and N. Doraswamy, "Server Cache Synchronization Protocol (SCSP)", RFC 2334, April 1998

[252] M.R. Macedonia, and D.P. Brutzman, "MBone Provides Audio and Video over the Internet", IEEE Computer, Vol. 27, No. 4, pp. 30-36, April 1994

[253] M. Maresca, N. Zingirian, and P. Baglietto, "Internet Protocol Support for Telephony", Proceedings of the IEEE, Vol. 92, No. 9, pp. 1463-1477, September 2004

[254] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam, "Assessing the Quality of Voice communications Over Internet Backbones", IEEE/ACM Transactions on Networking, Vol. 11, No. 5, pp. 747-760, October 2003

[255] K. Mase, "Toward Scalable Admission Control for VoIP Networks", IEEE Communications Magazine, Vol. 42, No. 7, pp. 42-47, July 2004

[256] S. McCanne, "vic: A Flexible Framework for Packet Video", Proceedings of the 3rd ACM International Conference on Multimedia, pp. 511-522, San Francisco, CA, November 1995

[257] S. McCanne, "Scalable Multimedia Communication Using IP Multicast and Lightweight Sessions", IEEE Internet Computing, Vol. 3, No. 2, pp. 33-45, March 1999

[258] D. McDysan, and D. Spohn, "ATM: Theory and Applications", October 1998

[259] D. McPherson, and K. Patel, "Experience with the BGP-4 Protocol", RFC 4277, January 2006

[260] C. McTaggart, "Telephone Numbers, Domain Names and ENUMbers", IEEE Communications Magazine, Vol. 40, No. 9, pp. 26, September 2002

[261] D.P. Mehta, and S. Sahni, "Handbook of Data Structures", Chapman and Hall/CRC, October 2004

[262] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "IPv4 Address Allocation and the BGP Routing Table Evolution", ACM SIGCOMM Computer Communication Review, Vol. 35, No. 1, pp. 71-80, January 2005

[263] R.M. Metcalfe, and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks", Communications of the ACM, Vol. 19, No. 5, pp. 395-404, July 1976

[264] D. Meyer, and K. Patel, "BGP-4 Protocol Analysis", RFC 4274, January 2006

[265] A. Milanovic, S. Srbljic, I. Raznjevic, D. Sladden, I. Matosevic, and D. Skrobo, "Methods for Lawful Interception in IP Telephony Networks Based on H.323", Proceedings of the IEEE EUROCON 2003, pp. 198-202, Ljubljana, September 2003

[266] B.A. Miller, T. Nixon, C. Tai, and M.D. Wood, "Home Networking with Universal Plug and Play", IEEE Communications Magazine, Vol. 39, No. 12, pp. 104-109, December 2001

[267] E. Miller, F. Andreasen, and G. Russell, "The PacketCable Architecture", IEEE Communications Magazine, Vol. 39, No. 6, pp. 90-96, June 2001

[268] D. Minoli, "Optimal Packet Length For Packet Voice Communication", IEEE Transactions on Communications, Vol. COM-27, No. 3, pp. 607-611, March 1979

[269] D. Minoli, "Voice over IPv6: Architectures for Next Generation VoIP Networks", Newnes Press, April 2006

[270] M. Mintz-Habib, A. Rawat, H. Schulzrinne, and X. Wu, "A VoIP Emergency Services Architecture and Prototype", Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN '05), pp. 523-528, San Diego, CA, October 2005

[271] K.D. Mitnick, and W.L. Simon, "The Art of Deception", Wiley, October 2003

[272] K.D. Mitnick, and W.L. Simon, "The Art of Intrusion", Wiley, March 2005

[273] P. V. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, November 1987

[274] W.A. Montgomery, "Techniques for Packet Voice Synchronization" IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No. 6, pp. 1022-1028, December 1983

[275] J. Moy, "OSPF Version 2", RFC 2328, April 1998

[276] R.R. Muduganti, S.K. Sogani, and H. Hexmoor, "Comparison of Information

Technology Adoption Rates Across Laggards, Innovators and Others", Proceedings of the 2005 IEEE International Conference on Information Technology: Coding and Computing (ITCC '05), Vol. 1, pp. 470-475, Las Vegas, NV, April 2005

[277] T. Murakami, M. Aihara, N. Shimamoto, and K. Ono, "SS7 Gateway for Inter-working between an IP Network and a Telephone Network", Proceedings of the joint IEEE 5th Asia-Pacific Conference on Communications and 4th Optoelectronics and Communications Conference (APCC/OECC '99), Vol. 1, pp. 57-60, Beijing, China, October 1999

[278] W. Naylor, and L. Kleinrock, "Stream Traffic Communication in Packet Switched Networks: Destination Buffering Considerations", IEEE Transactions on Communications, Vol. 30, No. 12, December 1982

[279] E. Noel, and K.W. Tang, "Performance Analysis of a VoIP Access Architecture", proceedings of the 2004 IEEE International Conference on Parallel Processing Workshops (ICPPW '04), pp. 282-290, August 2004

[280] L. Nuaymi, "WiMAX: Technology for the Last Mile", Wiley, August 2006

[281] F.D. Ohrtman, "Softswitch: Architecture for VoIP", McGraw-Hill, June 2006

[282] L. Ong, I. Rytina, M. Garcia, H.J. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, and C. Sharp, "Framework Architecture for Signalling Transport", RFC 2719, October 1999

[283] K. Ono, and S. Tachimoto, "End-to-middle Security in the Session Initiation Protocol (SIP)", Internet Draft, draft-ietf-sip-e2m-sec-01, October 2005

[284] J. Ott, C. Perkins, and D. Kutscher, "A Message Bus for Local Coordination", RFC 3259, April 2002

[285] F.J. Owens, "Signal Processing of Speech", MacMillan, March 1993

[286] C. Perkins, "RTP: Audio and Video for the Internet", Wiley, June 2002

[287] C. Perkins, and O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998

[288] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio", IEEE Network, Vol. 2, No. 5, pp. 40-48, September 1998

[289] R. Perlman, "Interconnections", Addison-Wesley, October 1999

[290] J. Peterson, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)", RFC 3853, July 2004

[291] J. Peterson, "ENUM Service Registration for Session Initiation Protocol (SIP)

Addresses-of-Record", RFC 3764, April 2004

[292] S. Petrack, and L. Conroy, "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services", RFC 2848, June 2000

[293] M. Pioro, and D. Medhi, "Routing, Flow and Capacity Design in Communication and Computer Networks", Elsevier, April 2004

[294] M. Poikselka, A. Niemi, and H. Khartabil, "The IMS: IP Multimedia Concepts and Services in the Mobile Domain", Wiley, January 2006

[295] C. Polyzois, K.H. Purdy, and P.-F. Yang, "From POTS to PANS: A Commentary on the Evolution to Internet Telephony", IEEE Internet Computing,

[296] J. D. Postel, "User Datagram Protocol", RFC 768, August 1980

[297] J. D. Postel, "Internet Protocol", RFC 791, September 1981

[298] J. D. Postel, "Transmission Control Protocol", RFC 793, September 1981

[299] L. Press, "Net.Speech: Audio Comes to the Net", Communications of the ACM, Vol. 38, No. 10, pp. 25-31, October 1995

[300] R. Puzmanova, "Routing and Switching", Pearson Education, December 2001

[301] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain Traffic Engineering with BGP", IEEE Communications Magazine, Vol. 41, No. 5, pp. 122-128, May 2003

[302] L. Rabiner, J. Cooley, H. Helms, L. Jackson, J. Kaiser, C. Rader, R. Schafer, K. Steiglitz, C. Weinstein, "Terminology in digital signal processing", IEEE Transactions on Audio and Electroacoustics, Vol. AU-20, No. 5, pp. 322-337, December 1972

[303] R. Ramjee, J.F. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive Playout Mechanisms for Packetised Audio Applications in Wide-Area Networks", Proceedings of IEEE INFOCOM '94, Vol. 2, pp. 680-688, June 1994

[304] B. Ramsdell (ed.), "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004

[305] J.F. Ransome, and J.W. Rittinghouse, "VoIP Security", Elsevier Digital Press, January 2005

[306] Y. Rekhter (ed.), T. Li (ed.), and S. Hares (ed.), "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006

[307] P. Resnick, and R. Virzi, "Skip and Scan: Cleaning Up Telephone Interfaces", Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '89), pp. 419-426, Monterey, CA, May 1992

[308] A. Rix, and M.P. M. Hollier, "The Perceptual Analysis Measurement System for Robust End-to-End Speech Quality Assessment", Proceedings of the IEEE ICASSP, p.p. 1515-1518, June 2000

[309] E.M. Rogers, "Diffusion of Innovations", Simon & Schuster International, November 2003

[310] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001

[311] J. Rosenberg, "The Real Time Transport Protocol (RTP) Denial of Service (DoS) Attack and its Prevention", Internet Draft, draft-rosenberg-mmusic-rtp-denialofservice-00, June 2003

[312] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", Internet Draft, draft-ietf-mmusic-ice-06, October 2005

[313] J. Rosenberg, R. Mahy, and C. Huitema, "Traversal Using Relay NAT (TURN)", Internet Draft, draft-rosenberg-midcom-turn-08, September 2005

[314] J. Rosenberg, H. Salama, and M. Squire, "Telephony Routing over IP (TRIP), RFC 3219, January 2002

[315] J. Rosenberg, and H. Schulzrinne, "Internet Telephony Gateway Location", Proceedings of IEEE INFOCOM '98, Vol. 2, pp. 488-496, March-April 1998

[316] J. Rosenberg, and H. Schulzrinne, "A Framework for Telephony Routing over IP ", RFC 2871, June 2000

[317] J. Rosenberg, and H. Schulzrinne, "Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP)", Internet Draft, draft-ietf-sip-guidelines-09, February 2005

[318] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002

[319] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003

[320] L. Roychoudhuri, E. Al-Shaer, H. Hamed, and G.B. Brewster, "Audio Transmission over the Internet: Experiments and Observations", Proceedings of the IEEE International Conference on Communications (ICC '03), pp. 552-556, Anchorage, AL, May 2003

[321] J. Ruiz, A. Vallejo, and J. Abella, "IPv6 conformance and interoperability testing",

Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005), pp. 83-88, June 2005

[322] P. Saint-Andre (ed.), "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004

[323] S. Salsano, L. Veltri, and D. Papalilo, "SIP Security Issues: the SIP Authentication Procedure and its Processing Load", IEEE Network, Vol. 16, No. 6, pp. 38-44, November-December 2002

[324] J.H. Saltzer, D.P. Reed, and D.D. Clark, "End-to-End Arguments in System Design", ACM Transactions in Computer Systems, Vol. 2, No. 4, pp. 277-288, November 1984

[325] P. Savage, "Designing a GUI for Business Telephone Users", ACM Interactions, Vol. 2, No. 1, pp. 32-41, January 1995

[326] J. Schiller, "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", RFC 3365, August 2002

[327] B. Schneier, "Secrets and Lies: Digital Security in a Networked World", Wiley, March 2004

[328] B. Schneier, M.T. Goodrich, and R. Tamassia, "Introduction to Security and Applied Cryptography", Wiley, October 2006

[329] C.-D. Schulz, "The MICE Project: Multimedia Integrated Conferencing for Europe (MICE)", Proceedings of the 2nd ACM International Conference on Multimedia (MM '94), Vol. 1, pp. 482-483, San Francisco, CA, October 1994

[330] H. Schulzrinne, "Voice Communication Across the Internet: A Network Voice Terminal", Department of Computer Science, University of Massachusetts, Technical Report, pp. 1-34, July 1992

[331] H. Schulzrinne, "Re-engineering the Telephone System", Proceedings of the IEEE Singapore International Conference on Networks (SICON)", Singapore, April 1997

[332] H. Schulzrinne, "The IETF Internet Telephony Architecture and Protocols", IEEE Network, Vol. 13, No. 3, pp. 18-23, May-June 1999

[333] H. Schulzrinne, and C. Agboh, "Session Initiation Protocol (SIP)-H.323 Interworking Requirements", RFC 4123, July 2005

[334] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998

[335] H. Schulzrinne, and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Proceedings of the 8th International Workshop on Network and Operating

System Support for Digital Audio and Video (NOSSDAV '98), Cambridge, UK, 1998

[336] H. Schulzrinne, and S. Petrack, "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 2833, May 2000

[337] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003

[338] M.C. Schlesener, and V.S. Frost, "Performance Evaluation of Telephony Routing over IP (TRIP)", Proceedings of the 3rd IEEE Workshop on IP Operations and Management (IPOM '03), pp. 47-53, Kansas City, MI, October 2003

[339] R. Seaford, "Secure Coding in C and C++", Addison-Wesley, September 2005

[340] S. Sharafeddine, A. Riedl, J. Glasmann, and J. Totzke, "On Traffic Characteristics and Bandwidth Requirements of Voice Over IP Applications", Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC 2003), Vol. 2, pp. 1324-1330, June-July 2003

[341] J. Shin, D.C. Lee, and C.-C. Jay Kuo, "Quality of Service for Internet Multimedia", Prentice Hall, July 2003

[342] R. Shirey, "Internet Security Glossary", RFC 2828, January 2000

[343] D.C. Sicker, and T. Lookabaugh, "VoIP Security: Not an Afterthought", ACM Queue, Vol. 2, No. 6, pp. 56-64, September 2004

[344] W. Simpson (ed.), "The Point-to-Point Protocol (PPP), RFC 1661, July 1994

[345] K. Singh, and H. Schulzrinne, "Interworking Between SIP/SDP and H.323", Proceedings of the 1st IP Telephony Workshop (IPTEL 2000), Berlin, April 2000

[346] K. Singh, and H. Schulzrinne, "Peer-to-Peer Internet Telephony Using SIP", Proceedings of the 15th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '05), Washington, DC, June 2005

[347] L. Slutsman (ed.), I. Faynberg, H. Lu, and M. Weissman, "The SPIRITS Architecture", RFC 3136, June 2001

[348] A.S. Spanias, "Speech Coding: A Tutorial Review", Proceedings of the IEEE, Vol. 82, No. 10, pp. 1541-1582, October 1994

[349] M. Squire, "A Gateway Location Protocol", Internet Draft, draft-ietf-iptel-glp-00, February 1999

[350] S. Sriram, "Lawful Intercept procedure via the Session Initiation Protocol (SIP) for the Open Mobile Alliance (OMA) Push to talk over Cellular (PoC)", Internet Draft, draft-sriram-sipping-poc-lip-01, August 2005

[351] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002

[352] Y. Stein, R. Shashoua, R. Insler, and M. Anavi, "TDM over IP", Internet Draft, draft-ietf-pwe3-tdmoip-04, September 2005

[353] W.R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols", February 1994

[354] J.W. Stewart III, "BGP4: Inter-Domain Routing in the Internet", December 1998

[355] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000

[356] A.J. Stienstra, "Technologies for DVB Services on the Internet", Proceedings of the IEEE, Vol. 94, No. 1, pp. 228-236, January 2006

[357] I. Stoica, "Stateless Core: A Scalable Approach for Quality of Service in the Internet", Springer-Verlag, May 2004

[358] T. Stone, R. Alena, and M. Johnson, "IP Telephony for Interplanetary Exploration", Proceedings of the 2004 IEEE Aerospace Conference, Vol. 2, pp. 1217-1230, Big Sky, MT, March 2004

[359] M. Streenstrup, "Routing in Communications Networks", Prentice Hall, May 1995

[360] M. Stukas, and D.C. Sicker, "An Evaluation of VoIP Traversal of Firewalls and NATs Within an Enterprise Environment", Iformation Systems Frontiers, Vol. 6, No. 3, pp 219-228, September 2004

[361] L. Subramanian, S. Agarwal, J. Rexford, and R.H. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points", Proceedings of IEEE INFOCOM '02, Vol. 2, pp. 618-627, June 2002

[362] R.P. Swale, P. A. Mart, P. Sijben, S. Brim, and M. Shore, "Middlebox Communications (midcom) Protocol Requirements", RFC 3304, August 2002

[363] A. Tanenbaum, "Computer Networks", Pearson Education, August 2002

[364] D. Terzis, "Frame Relay Networks: Interoperability with ATM and Voice Transportation", British Telecom, BT Labs Technical Report, Ipswitch, UK, October 1996

[365] D. Terzis, "A Standards-based Java ATM Library for Linux", 6th International Linux Kongress, Augsburg, Germany, September 1999

[366] D. Terzis, "VIA Project Report", Nortel, Harlow Labs Technical Report, Harlow, UK, March 2000

[367] D. Terzis, J. Crowcroft, and J. Cable, "End-to-end Call Signaling Between GSTN and IP Networks", Proceedings of the IEE 16th UK Teletraffic Symposium, pp/ 35/1-35/10, Harlow, UK, May 2000

[368] D. Terzis, "An Alternative Framework for IP Telephony Call Routing (ACRF)", Nortel Design Forum, Harlow, UK, December 2000

[369] C. Topolcic, "Experimental Internet Stream Protocol, Version 2", RFC 1190, October 1990

[370] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer 2 Tunneling Protocol (L2TP)", RFC 2661, August 1999

[371] P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP", RFC 3065, February 2001

[372] T. Turletti, "The INRIA Videoconferencing System (IVS)", ConneXions - The Interoperability Report, Vol. 8, No. 10, pp. 20-24, October 1994

[373] M.E. Ulug, and J.G. Gruber, "Statistical multiplexing of data and encoded voice in a transparent intelligent network", Proceedings of the 5th Symposium on Data Communications, pp. 14-20, September 1977

[374] H. Uzunalioglu, K.P. Das, and D.J. Houck, "An Architecture and Admission Control Algorithm for Multi-Precedence Voice over IP (VoIP) Calls", Proceedings of the IEEE Military Communications Conference (MILCOM '04), pp. 906-912, October 2004

[375] A. Vaha-Sipila, "URLs for Telephony Calls", RFC 2806, April 2000

[376] A. Valencia, and T. Kollar, "Cisco Layer Two Forwarding (Protocol) (L2F)", RFC 2341, May 1998

[377] U. Varshney, A. Snow, M. McGivern, and C. Howard, "Voice over IP", Communications of the ACM, Vol. 45, No. 1, pp. 89-96, January 2001

[378] A. Vemuri, and J. Peterson, "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", RFC 3372, September 2002

[379] Q. Vohra, and E. Chen, "BGP Support for Four-octet AS Number Space", Internet Draft, draft-ietf-idr-as4bytes-12, November 2005

[380] S. Vuong, and Y. Bai, "A Survey of VoIP Intrusions and Intrusion Detection Systems", Proceedings of the IEEE 6th International Conference on Advanced Communication Technology (ICACT '04), Vol. 1, pp. 317-322, Gangwon-Do, Republic of Korea, February 2004

[381] W3C, Extensible Markup Language (XML) 1.1, Specification REC-xml11-20040204,

February 2004

[382] K. Wacks, "Home Systems Standards: Achievements and Challenges", IEEE Communications Magazine, Vol. 40, No. 4, pp. 152-159, April 2002

[383] B.W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet", Proceedings of the International Symposium on Multimedia Software Engineering, pp. 17-24, December 2000

[384] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997

[385] T. Wallingford, "Switching to VoIP", O'Reilly, July 2005

[386] T.J. Walsh, and D.R. Kuhn, "Challenges in Securing Voice over IP", IEEE Security & Privacy Magazine, Vol. 3, No. 3, pp. 44-49, May-June 2005

[387] S. Wang, L. Nieto, and E. Zielinski, "A Study of IP Network Impairments Impact on Media Gateway Performance", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '01), Vol. 4, pp. 2596-2600, November 2001

[388] Z. Wang, "Internet QoS: Architectures and Mechanisms", Morgan Kaufmann, October 2000

[389] C.J. Weinstein, and J.W. Forgie, "Experience with speech communication in packet networks", IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No. 6, pp. 963-980, December 1983

[390] J. Wisely, P. Eardley, and L. Burness, "IP for 3G", June 2002

[391] S. Williams, "The Softswitch Advantage", IEE Review, Vol. 48, No. 4, pp. 25-29, July 2002

[392] R. Wittmann, and M. Zitterbart, "Multicast Communication - Protocols, Programming and Applications", Academic Press, June 2000

[393] D.J. Wright, "Voice over ATM: An Evaluation of Implementation Alternatives", IEEE Communications Magazine, Vol. 34, No. 5, pp. 72-80, May 1996

[394] D.J. Wright, "Voice over Packet Networks", Wiley, February 2001

[395] S. Wright, and R. Onvuvar, "IP "Telephony" vs. ATML What is There to Discuss?", Proceedings of the 1st IEEE International Conference on ATM (ICATM '98), pp. 400-409, June 1998

[396] K. Xu, A.-P. Wang, B. Wang, and J.-P. Wu, "NLRI Storage Aggregation of BGP-4 Routing Information", Proceedings of the 10th IEEE International Conference on Networks (ICON 2002), pp. 57-62, Singapore, August 2002

[397] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000

[398] N. Yin, S. Li, and T.E. Stern, "Congestion Control for Packet Voice by Selective Packet Discarding, IEEE Transactions on Communications, Vol. COM-38, No. 5, pp. 674-683, May 1990

[399] H.-K. Yoo, and B.-R. Kang, "A Media Stream Processing of VoIP Media Gateway", Proceedings of the IEEE 9th Asia-Pacific Conference on Communications (APCC '03), Vol. 1, pp. 91-94, Penang, Malaysia, September 2003

[400] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the Internet AS-Level Topology", ACM SIGCOMM Computer Communication Review, Vol. 35, No. 1, pp. 53-61, January 2005

[401] Y. Zhang, "SIP-based VoIP Network and its Interworking with the PSTN", IEE Electronics & Communication Engineering Journal, Vol. 14, No. 6, pp. 273-282, December 2002

R2. Online Resources

[U1] Asterisk, The Open Source PBX, http://www.asterisk.org/

[U2] British Telecom, http://www.bt.com/

[U3] Brix Networks, TestYourVoIP.com, http://www.testyourvoip.com/

[U4] BroadVoice, http://www.broadvoice.com/

[U5] CableLabs, http://www.cablelabs.com/

[U6] CableModem (DOCSIS), http://www.cablemodem.com/

[U7] Cisco Systems, http://www.cisco.com/

[U8] eTForecasts, http://www.etforecasts.com/

[U9] European Telecommunications Standards Institute (ETSI), http://www.etsi.org/

[U10] Federal Communications Commission (FCC), http://www.fcc.gov/

[U11] C. Feather, "The BT Network", http://www.davros.org/

[U12] Gizmo, http://www.gizmoproject.com/

[U13] Google Talk, http://www.google.com/talk/

[U14] HomeChoice, http://www.homechoice.co.uk/

[U15] G. Huston, "BGP Reports", http://bgp.potaroo.net/index-bgp.html

[U16] Information Sciences Institute (ISI), http://www.isi.edu/

[U17] Internet Engineering Task Force (IETF), http://www.ietf.org/

[U18] Internet2, http://www.internet2.edu/

[U19] Internet2 VoIP Working Group, http://voip.internet2.edu/

[U20] International Numbering Plans, http://www.numberingplans.com/

[U21] International Telecommunication Union (ITU), http://www.itu.int/

[U22] Internet Telephony Products, http://www.iptel.org/info/products/

[U23] IP Telephony Resources, http://www.iptelephony.org/

[U24] Jabber, http://www.jabber.com/

[U25] KaZaA, http://www.kazaa.com/

[U26] Lawrence Berkeley National Laboratory (LNBL), http://www.lbnl.org/

[U27] Lucent, http://www.lucent.com/

[U28] Microsoft, http://www.microsoft.com/

[U29] Microsoft NetMeeting, http://www.microsoft.com/windows/netmeeting/

[U30] Motorola, http://www.motorola.com/

[U31] Narus, http://www.narus.com/

[U32] Net2Phone, http://www.net2phone.com/

[U33] Nortel, http://www.nortel.com/

[U34] OpenH323 Project, http://www.openh323.org/

[U35] OpenH323 Gatekeeper, http://www.gnugk.org/

[U36] Open OSP Project, http://www.openosp.org/

[U37] PacketCable, http://www.packetcable.com/

[U38] PacPhone, http://www.pacphone.com/

[U39] Pulver.com, http://www.pulver.com/

[U40] RAD Data Communications, http://www.rad.com/

[U41] RealAudio, http://www.real.com/

[U42] Ridgeway Systems, http://www.ridgewaysystems.com (now part of Tandberg, http://www.tandberg.net/)

[U43] Robust Audio Tool (RAT), RAT Website, http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[U44] Routeviews, http://www.routeviews.org/

[U45] SIP Express Router (SER), http://www.iptel.org/ser/

[U46] SIP Servlet API, http://www.sipservlet.org/

[U47] Sky, http://www.sky.com/

[U48] Skype, http://www.skype.com/

[U49] Speak Freely, http://www.speakfreely.org/

[U50] Speex, http://www.speex.org/

[U51] Sun Microsystems, Java Media Framework API (JMF), http://java.sun.com/products/java-media/jmf/

[U52] TDMoIP, http://www.tdmoip.com/

[U53] TeleGeography Research, "TeleGeography 2006 Report Executive Summary", http://www.telegeography.com/ee/free_resources/pdf/tg2006_exec_summ.pdf

[U54] Third Generation Partnership Project (3GPP), http://www.3gpp.org/

[U55] Third Generation Partnership Project 2 (3GPP2), http://www.3gpp2.org/

[U56] UCL-CS (Department of Computer Science, University College London), http://www.cs.ucl.ac.uk/

[U57] U.S. Census Bureau, "World Population Information", http://www.census.gov/ipc/www/world.html

[U58] Visual Audio Tool (vat), http://www-nrg.ee.lbl.gov/vat/

[U59] VocalTec Communications, http://www.vocaltec.com/

[U60] voip-info.org, http://www.voip-info.org/

[U61] VoIP Security Alliance (VOIPSA), http://www.voipsa.org/

[U62] VoIP Security Blog, http://www.voip-security-blog.com/

[U63] VOMIT, Voice over Misconfigured Internet Telephones, http://vomit.xtdnet.nl/

[U64] VoipBuster, http://www.voipbuster.com/

[U65] Vonage, http://www.vonage.com/

[U66] Vovida, http://www.vovida.org/

[U67] Windows Live Messenger, http://get.live.com/messenger/overview/

[U68] Yahoo! Messenger, http://messenger.yahoo.com/

<u>NOTE</u>: The validity of the above URLs has been last confirmed on 22 October 2006.

APPENDICES

A: Using JMF to Create an Audio Tool

Following is an example, with full source code, of how the media engine of an audio tool can be developed from the Java Media Framework (JMF) libraries [U51].

```
package audiotool;

import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.net.*;
import java.util.*;
import javax.media.*;
import javax.media.control.*;
import javax.media.format.*;
import javax.media.protocol.*;
import javax.media.rtp.*;
import javax.media.rtp.event.*;

import javax.media.rtp.rtcp.*;

import com.sun.media.rtp.*;


public class AudioTool implements Runnable, SessionListener,
SendStreamListener, ReceiveStreamListener, RemoteListener,
ControllerListener
{
    private static boolean debuggingMode = true;

    private String destinationAddress;
    private int destinationPort;
    private SessionManager manager;
    private CaptureDeviceInfo captureDevice;
    private Processor processor;      // Processes the SendStream
    private Player player;         // Processes the ReceiveStream

    private String url;
    private MediaLocator mrl;

    public AudioTool(String address, int port)
    {
        this.destinationAddress = address;
        this.destinationPort = port;

        if ((address != null) && (port > 1024) && (port < 65535) &&
        (port%2==0))
        {
            this.start();
        }
        else
        {
            System.out.print("\nInvalid IP address and/or port, cannot
                                run AudioTool");
            System.exit(-1);
        }
    }
```

```
public void start()
{
    debug("AudioTool initialising...");

    if ((captureDevice=createCaptureDevice()) != null)
    {
        debug("\n\nDetected capture device " + captureDevice);
        mrl = captureDevice.getLocator();
    }
    else
    {
        debug("\nCould not detect any capture devices. Program
           terminated.");
        System.exit(-1);
    }

    debug("\n\nCreating Processor for MediaLocator " + mrl);

    if ((processor=createProcessor(mrl)) == null)
    {
        debug("\nCould not create media processor for destination"
           + this.destinationAddress + ":" + this.destinationPort
           + ". Program terminated.");
        System.exit(-1);
    }

    if ((manager=createManager(this.destinationAddress,
    this.destinationPort)) == null)
    {
        debug("\nCould not create RTP SessionManager for
              destination " + this.destinationAddress + ":" +
              this.destinationPort + ". Program terminated.");
        System.exit(-1);
    }

    this.addListeners(this.manager);

    try
    {
        DataSource ds = processor.getDataOutput();
        debug("\nDataSource " + ds + ", content type \"" +
              ds.getContentType() + "\"");

        SendStream ss = createSendStream(manager, ds, 0);

        ss.start();
    }
    catch(Throwable t)
    {
        debug("\nCould not create SendStream.\n\nThe following
              error occurred: " + t);
    }

    processor.start();
}

public void run()
{
    debug("\nAudio Gateway running...");
}
```

```
// ================ Core functionality ====================== //

public SessionManager createManager(String address, int port)
{
    int ttl = 1;
    SessionManager mgr;

    mgr = (SessionManager)new com.sun.media.rtp.RTPSessionMgr();

    if (mgr == null)
        return null;

    // RTP Canonical Name (CNAME) of the local participant
    String cname = mgr.generateCNAME();

    SourceDescription[] sourceDescriptionList =
         this.getSourceDescriptionList();

    // Set the CNAME
    sourceDescriptionList[1].setDescription(cname);

    // Local Session Address
    SessionAddress localSessionAddress = new SessionAddress();

    try
    {
        InetAddress destinationAddress =
                InetAddress.getByName(address);

        SessionAddress remoteSessionAddress  = new
                SessionAddress(destinationAddress, port,
                destinationAddress, port+1);

        mgr.initSession(localSessionAddress,
                sourceDescriptionList, 0.05, 0.25);

        mgr.startSession(remoteSessionAddress, ttl, null);
    }
    catch (Exception e)
    {
        debug("\nError initializing RTP Manager: " +
                e.getMessage());
        return null;
    }

    return mgr;
}

public static SourceDescription[]
     createSourceDescriptionList(String name, String cname,
     String email)
{
    return createSourceDescriptionList(name, cname, email, "", "",
     "", "", "");
}

public static SourceDescription[]
     createSourceDescriptionList(String name, String cname,
     String email, String phone, String location,
     String tool, String privateData, String description)
{
    SourceDescription[] sourceDescriptionList =
```

```
    {
        new SourceDescription(SourceDescription.SOURCE_DESC_NAME,
                name, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_CNAME,
                cname, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_EMAIL,
                email, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_PHONE,
                phone, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_LOC,
                location, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_TOOL,
                tool, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_PRIV,
                privateData, 1, false),
        new SourceDescription(SourceDescription.SOURCE_DESC_NOTE,
                description, 1, false)
    };

    return sourceDescriptionList;
}

public boolean addListeners(SessionManager mgr)
{
    if (mgr == null)
        return false;

    // mgr.addSessionListener(this);
    // mgr.addRemoteListener(this);
    // mgr.addSendStreamListener(this);
    mgr.addReceiveStreamListener(this);

    return true;
}

public CaptureDeviceInfo createCaptureDevice()
{
    AudioFormat format;
    CaptureDeviceInfo cdi;
    Vector devices;

    // Set the desired audio format (as supported by JMF,
    // reported in JMF Registry)
    format = new AudioFormat(AudioFormat.LINEAR);

    // Find a capture device that will record audio data
    // for this format
    devices = CaptureDeviceManager.getDeviceList(format);

    if ((devices != null) && (devices.size() > 0))
    {
        cdi = (CaptureDeviceInfo)devices.elementAt( 0);
        debug("\n\nDetected capture device " + cdi);
    }
    else
    {
        debug("\nCould not detect any capture devices!");
        return null;
    }

    return cdi;
}
```

```
public Processor createProcessor(Object source)
{
    Processor proc = null;

    try
    {
        if (source == null)
            proc = Manager.createProcessor(mrl);
        else if (source instanceof DataSource)
            proc = Manager.createProcessor((DataSource)source);
        else if (source instanceof MediaLocator)
            proc = Manager.createProcessor((MediaLocator)source);
        else if (source instanceof CaptureDeviceInfo)
            proc = Manager.createProcessor((MediaLocator)
                    ((CaptureDeviceInfo)source).getLocator());
        else
            return null;

        debug("\n" + proc + " created succesfully");
    }
    catch (Exception e)
    {
        debug("\n\nProcessor creation error: " + e);
    }

    debug("\n\nConfiguring " + proc + ", please wait\n...");

    do  // Block until the processor has been configured
    {
        debug(".");

        try
        {
            Thread.currentThread().sleep(1000);
        }
        catch (InterruptedException ie)
        {
        }

        // Configure the processor
        proc.configure();
    }
    while (proc.getState() != proc.Configured);

    debug("\n\nProcessor " + proc + " configured\n\n");

    proc.setContentDescriptor(new
        ContentDescriptor(ContentDescriptor.RAW_RTP));

    TrackControl track[] = proc.getTrackControls();
    boolean encodingOk = false;

    // Attempt to program one of the tracks for RTP data output
    for (int i = 0; i < track.length; i++)
    {
        if (!encodingOk && track[i] instanceof FormatControl)
        {
            if (((FormatControl)track[i]).setFormat(new
                  AudioFormat(AudioFormat.ULAW_RTP)) == null)
            {
                track[i].setEnabled(false);
```

```
            }
            else
            {
                encodingOk = true;
            }
        }
        else
        {
            // If the track can't be programmed for RTP,
            // it should be disabled
            track[i].setEnabled(false);
        }
    }

    debug("\n\nMedia tracks programmed");

    // If we can send RTP data, reallize the processor
    if (encodingOk)
    {
        debug("\n\nRealising " + proc + ", please wait\n...");

        do  // Block until the processor has been realized.
        {
            debug(".");

            try
            {
                Thread.currentThread().sleep(1000);
            }
            catch (InterruptedException ie)
            {
            }

            // Realize the processor
            proc.realize();
        }
        while (proc.getState() != proc.Realized);

        debug("\n\nProcessor " + proc + " realised\n\n");
    }

    return proc;
}

public SendStream createSendStream(SessionManager mgr, DataSource
    source, int streamIndex)
{
    SendStream stream = null;

    try
    {
        stream = mgr.createSendStream(source, streamIndex);
        debug("\nCreated SendStream " + stream);
    }
    catch (Throwable t)
    {
        debug("\nSendStream error: " + t);
    }

    return stream;
}
```

```
// ================== Event handlers ====================== //

public void update(SessionEvent event)
{
    Participant part = null;
    ReceiveStream stream = null;
    SessionManager sessionManager;
    Vector sendStreams = null;
    Vector receiveStreams = null;

    debug("\n\nNew SessionEvent: " + event);

    sessionManager = event.getSessionManager();

    if (sessionManager != null)
    {
        sendStreams = sessionManager.getSendStreams();
        receiveStreams = sessionManager.getReceiveStreams();

        debug("\n\nSessionManager: " + sessionManager);

        if (sessionManager == manager)
            debug("\n(identical to locally created Session
                      Manager)");
    }
    else
    {
        debug("\n\nNo SessionManager object");
    }

    if (sendStreams != null)
    {
        debug("\n\n" + sendStreams.size() + " send stream(s)");
    }
    else
    {
        debug("\n\nNo send streams found");
    }

    if (receiveStreams != null)
    {
        debug("\n\n" + receiveStreams.size() + " receive
        stream(s)");
    }
    else
    {
        debug("\n\nNo receive streams found");
    }

    // if this is a new participant event, add this participant to
    // the list of already existant participants.
    if (event instanceof NewParticipantEvent)
    {
        part = ((NewParticipantEvent)event).getParticipant();

        if (part == null)
            return;

        debug("\n\nNew session participant: " + part);
    }
}
```

```
public void update(ReceiveStreamEvent event)
{
    SessionManager mgr = (SessionManager)event.getSource();

    debug("\n\nNew ReceiveStreamEvent arrived");

    if (mgr == null)
        return;

    debug("\n\n" + mgr.getReceiveStreams().size() + "
           ReceiveStreams found");

    if (event instanceof NewReceiveStreamEvent)
    {
        if (this.player != null)
            return;

        String cname = "RTP Player";
        Participant part = null;
        DataSource source;
        ReceiveStream stream = null;
        Processor proc = null;

        try
        {
            stream =
                ((NewReceiveStreamEvent)event).getReceiveStream();
            part = stream.getParticipant();

            if (part != null)
            {
                cname = part.getCNAME();
            }

            debug("\n\nReceiveStream is " + stream);
            debug("\nParticipant is " + part);

            source = stream.getDataSource();

            debug("\n\nInput DataSource is " + source);

            if (source != null)
            {
                // We could also use a clone of source
                this.player = Manager.createPlayer(source);

                // new PlayerWindow(player, "Receiving...");

                player.start();
            }
        }
        catch (Exception e)
        {
            System.err.println("\n\nNewReceiveStreamEvent
                exception: " + e.getMessage() + "\n\n");
            return;
        }
    }
}
```

```
public void update(SendStreamEvent event)
{
    debug("\n\nNew SendStreamEvent: " +  event);
}


public void update(RemoteEvent event)
{
    Participant part = null;

    debug("\n\nNew RemoteEvent: " + event);

    if (event instanceof ReceiverReportEvent)
    {
        part =
        ((ReceiverReportEvent)event).getReport().getParticipant();
    }
    else if (event instanceof SenderReportEvent)
    {
        part =
        ((SenderReportEvent)event).getReport().getParticipant();
    }

    debug("\n\nNew remote participant: " + part);
}

public synchronized void controllerUpdate(ControllerEvent event)
{
    debug("\n\nController update event received: " + event);

    if (event instanceof RealizeCompleteEvent)
    {
        debug("\n\nPlayer " + event.getSourceController() + "
        realized");
    }
    else if ((event instanceof ControllerClosedEvent) ||
        (event instanceof ControllerErrorEvent)  ||
        (event instanceof DeallocateEvent))
    {
        Player p = (Player)event.getSourceController();
    }
}
```

```
    // ================= Convenience methods ===================== //

    // Returns a SourceDescription list; CNAME must be updated
    // after calling the method
    private SourceDescription[] getSourceDescriptionList()
    {
        SourceDescription[] sourceDescriptionList;
        String username = "LocalParticipant@" +
                          this.destinationAddress + ":" +
                          this.destinationPort;

        try
        {
            username = System.getProperty("user.name");
        }
        catch (SecurityException e)
        {
        }

        sourceDescriptionList = new SourceDescription[]
        {
            new SourceDescription(SourceDescription.SOURCE_DESC_NAME,
                 "Administrator", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_CNAME,
                 null, 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_EMAIL,
                 "user@synergy-tv.com", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_PHONE,
                 "01372-xxx xxx", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_LOC,
                 "Leeds", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_TOOL,
                 "Audio Transmitter v0.1", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_PRIV,
                 "********", 1, false),
            new SourceDescription(SourceDescription.SOURCE_DESC_NOTE,
                 "Excellent!", 1, false)
        };

        return sourceDescriptionList;
    }

    public static void setDebuggingMode(boolean newDebuggingMode)
    {
        AudioTool.debuggingMode = newDebuggingMode;
    }

    public static boolean getDebuggingMode()
    {
        return AudioTool.debuggingMode;
    }

    public static void debug(String message)
    {
        if (AudioTool.debuggingMode == true)
        {
            System.out.println(message);
        }
    }
}
```

B: The HITRA Class Hierarchy

The HITRA class hierarchy (tree format), as produced by Javadoc.

Package Hierarchies:

SSF.OS.HITRA, SSF.OS.HITRA.event.fsm, SSF.OS.HITRA.event.message, SSF.OS.HITRA.event.message.open, SSF.OS.HITRA.event.message.update, SSF.OS.HITRA.net, SSF.OS.HITRA.net.trib, SSF.OS.HITRA.test, SSF.OS.HITRA.timing, SSF.OS.HITRA.util

---

Class Hierarchy:

- class java.lang.Object
  - class SSF.OS.HITRA.event.message.open.**CapabilityInformation**
  - class SSF.OS.HITRA.event.message.open.**CapabilityInformation.RouteType**
  - class SSF.OS.HITRA.util.**Debug**
  - class SSF.OS.HITRA.net.**DecisionProcess**
  - class com.renesys.raceway.SSF.Entity
    - class SSF.OS.Timer
      - class SSF.OS.HITRA.timing.**PeerStartTimer**
      - class SSF.OS.HITRA.timing.**Timer**
        - class SSF.OS.HITRA.timing.**EventTimer**
  - class SSF.OS.HITRA.net.**Flooder**
  - class SSF.OS.HITRA.**Global**
  - class SSF.OS.HITRA.**HITRA**
  - class SSF.OS.HITRA.util.**IPaddress**
  - class SSF.OS.HITRA.util.**ObjCache**
  - class SSF.OS.HITRA.event.message.open.**Open.OptionalParam**
  - class SSF.OS.HITRA.event.message.update.**PathSegment**
  - class SSF.OS.HITRA.event.fsm.**PeerSession**
  - class SSF.OS.ProtocolMessage
    - class SSF.OS.HITRA.event.message.**Message**
      - class SSF.OS.HITRA.event.message.**Keepalive**
      - class SSF.OS.HITRA.event.message.**Notification**
      - class SSF.OS.HITRA.event.message.open.**Open**
      - class SSF.OS.HITRA.event.message.**StartStop**
      - class SSF.OS.HITRA.event.message.**Timeout**
      - class SSF.OS.HITRA.event.message.**Transport**
      - class SSF.OS.HITRA.event.message.update.**Update**
  - class SSF.OS.ProtocolSession (implements com.renesys.raceway.DML.Configurable)
    - class SSF.OS.HITRA.test.**SSFConfigTest**
    - class SSF.OS.HITRA.test.**SSFMessagingTest**
    - class SSF.OS.HITRA.**TRIPSession**
  - class SSF.OS.HITRA.net.**Route**

- class SSF.OS.HITRA.event.message.update.**RoutingAttribute**
  - class SSF.OS.HITRA.event.message.update.**AdvertisementPath**
  - class SSF.OS.HITRA.event.message.update.**AtomicAggregate**
  - class SSF.OS.HITRA.event.message.update.**ITADTopology**
  - class SSF.OS.HITRA.event.message.update.**LocalPreference**
  - class SSF.OS.HITRA.event.message.update.**MultiExitDisc**
  - class SSF.OS.HITRA.event.message.update.**NextHopServer**
  - class SSF.OS.HITRA.event.message.update.**ReachableRoutes**
  - class SSF.OS.HITRA.event.message.update.**RoutedPath**
  - class SSF.OS.HITRA.event.message.update.**WithdrawnRoutes**
- class SSF.OS.HITRA.net.**RoutingObject**
- class SSF.OS.HITRA.net.trib.**RoutingTable**
- class SSF.OS.HITRA.util.**StringManip**
- class java.lang.Throwable (implements java.io.Serializable)
  - class java.lang.Exception
    - class SSF.OS.ProtocolException
      - class SSF.OS.HITRA.**TRIPException**
- class SSF.OS.HITRA.net.trib.**TRIB**
  - class SSF.OS.HITRA.net.trib.**AdjTRIBIn**
  - class SSF.OS.HITRA.net.trib.**AdjTRIBOut**
  - class SSF.OS.HITRA.net.trib.**ExtTRIB**
  - class SSF.OS.HITRA.net.trib.**LocalRoutes**
  - class SSF.OS.HITRA.net.trib.**LocTRIB**
- class SSF.OS.HITRA.util.**Trie**
- class SSF.OS.HITRA.util.**Trie.Node**
- class SSF.OS.HITRA.event.fsm.**TRIPState** (implements SSF.OS.HITRA.event.fsm.TRIPEventsInterface)
  - class SSF.OS.HITRA.event.fsm.**Active**
  - class SSF.OS.HITRA.event.fsm.**Connect**
  - class SSF.OS.HITRA.event.fsm.**Established**
  - class SSF.OS.HITRA.event.fsm.**Idle**
  - class SSF.OS.HITRA.event.fsm.**OpenConfirm**
  - class SSF.OS.HITRA.event.fsm.**OpenSent**
- class SSF.OS.HITRA.util.**UniqueIDFactory**
- class SSF.OS.HITRA.event.message.update.**Update.AscendingSorter** (implements java.util.Comparator)
- class SSF.OS.HITRA.util.**WeightedInBuffer**

Interface Hierarchy

- interface SSF.OS.HITRA.event.fsm.**TRIPEventsInterface**